Attention on Attention for Image Captioning

Lun Huang[1] Wenmin Wang[1,3*] Jie Chen[1,2] Xiao-Yong Wei[2]
[1]School of Electronic and Computer Engineering, Peking University
[2]Peng Cheng Laboratory
[3]Macau University of Science and Technology
huanglun@pku.edu.cn, {wangwm@ece.pku.edu.cn, wmwang@must.edu.mo}, {chenj, weixy}@pcl.ac.cn

Abstract

*Attention mechanisms are widely used in current encoder/decoder frameworks of image captioning, where a weighted average on encoded vectors is generated at each time step to guide the caption decoding process. However, the decoder has little idea of whether or how well the attended vector and the given attention query are related, which could make the decoder give misled results. In this paper, we propose an "Attention on Attention" (AoA) module, which extends the conventional attention mechanisms to determine the relevance between attention results and queries. AoA first generates an "**information vector**" and an "**attention gate**" using the attention result and the current context, then adds another attention by applying element-wise multiplication to them and finally obtains the "**attended information**", the expected useful knowledge. We apply AoA to both the encoder and the decoder of our image captioning model, which we name as AoA Network (AoANet). Experiments show that AoANet outperforms all previously published methods and achieves a new state-of-the-art performance of 129.8 CIDEr-D score on MS COCO "Karpathy" offline test split and 129.6 CIDEr-D (C40) score on the official online testing server. Code is available at* `https://github.com/husthuaan/AoANet`.

1. Introduction

Image captioning is one of the primary goals of computer vision which aims to automatically generate natural descriptions for images. It requires not only to recognize salient objects in an image, understand their interactions, but also to verbalize them using natural language, which makes itself very challenging [25, 45, 28, 12].

Inspired by the development of neural machine translation, attention mechanisms have been widely used in current encoder/decoder frameworks for visual captioning [42, 27, 47, 2, 7, 16, 15] and achieved impressive results. In such a framework for image captioning, an image is first encoded to a set of feature vectors via a CNN based network and then decoded to words via an RNN based network, where the attention mechanism guides the decoding process by generating a weighted average over the extracted feature vectors for each time step.

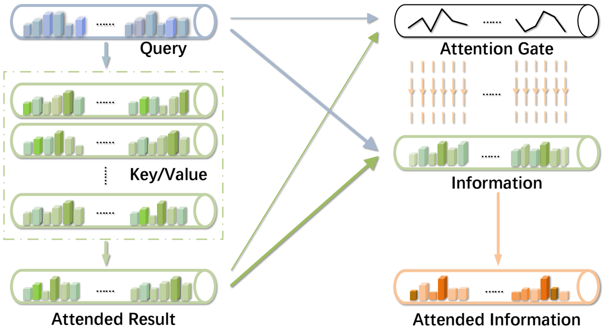


Figure 1: *Attention on Attention* (AoA). AoA generates an **information vector** and an **attention gate** using the attention result and the attention query, and adds another attention by applying the gate to the information and obtains the **attended information**.

The attention mechanism plays a crucial role in such a system that must capture global dependencies, *e.g.* a model for the sequence to sequence learning task like image/video captioning, since the output is directly conditioned on the attention result. However, the decoder has little idea of whether or how well the attention result is related to the query. There are some cases when the attention result is not what the decoder expects and the decoder can be misled to give fallacious results, which could happen when the attention module doesn't do well on its part or there's no worthful information from the candidate vectors at all. The former case can't be avoided since mistakes always happen. As for the latter, when there's nothing that meets the

[*] Corresponding author

requirement of a specific query, the attention module still returns a vector which is a weighted average on the candidate vectors and thus is totally irrelevant to the query.

To address this issue, we propose *Attention on Attention* (AoA), which extends the conventional attention mechanisms by adding another attention. Firstly, AoA generates an "**information vector**" and an "**attention gate**" with two linear transformations, which is similar to GLU [10]. The information vector is derived from the current context (*i.e.* the query) and the attention result via a linear transformation, and stores the newly obtained information from the attention result together with the information from the current context. The attention gate is also derived from the query and the attention result via another linear transformation with sigmoid activation followed, and the value of each channel indicates the relevance/importance of the information on the corresponding channel in the information vector. Subsequently, AoA adds another attention by applying the attention gate to the information vector using element-wise multiplication and finally obtains the "**attended information**", the expected useful knowledge.

AoA can be applied to various attention mechanisms. For the traditional single-head attention, AoA helps to determine the relevance between the attention result and query. Specially, for the recently proposed multi-head attention [35], AoA helps to build relationships among different attention heads, filters all the attention results and keeps only the useful ones.

We apply AoA to both the image encoder and the caption decoder of our image captioning model, AoANet. For the encoder, it extracts feature vectors of objects in the image, applies self-attention [35] to the vectors to model relationships among the objects, and then applies AoA to determine how they are related to each other. For the decoder, it applies AoA to filter out the irrelevant/misleading attention results and keep only the useful ones.

We evaluate the impact of applying AoA to the encoder and decoder respectively. Both quantitative and qualitative results show that AoA module is effective. The proposed AoANet outperforms all previously published image captioning models: a single model of AoANet achieves 129.8 CIDEr-D score on MS COCO dataset offline test split; and an ensemble of 4 models achieves 129.6 CIDEr-D (C40) score on the online testing server. Main contributions of this paper include:

- We propose the *Attention on Attention* (AoA) module, an extension to the conventional attention mechanism, to determine the relevance of attention results.
- We apply AoA to both the encoder and decoder to constitute AoANet: in the encoder, AoA helps to better model relationships among different objects in the image; in the decoder, AoA filters out irrelative attention results and keeps only the useful ones.
- Our method achieves a new state-of-the-art performance on MS COCO dataset.

2. Related Work

2.1. Image Captioning

Earlier approaches to image captioning are rule/template-based [48, 34] which generate slotted caption templates and use the outputs of object detection [30, 39, 38], attribute prediction and scene recognition to fill in the slots. Recent approaches are neural-based and specifically, utilize a deep encoder decoder framework, which is inspired by the development of neural machine translation [8]. For instance, an end-to-end framework is proposed with a CNN encoding the image to feature vector and an LSTM decoding it to caption [37]. In [42], the spatial attention mechanism on CNN feature map is used to incorporate visual context. In [6], a spatial and channel-wise attention model is proposed. In [27], an adaptive attention mechanism is introduced to decide when to activate the visual attention. More recently, more complex information such as objects, attributes and relationships are integrated to generate better descriptions [50, 2, 49, 44].

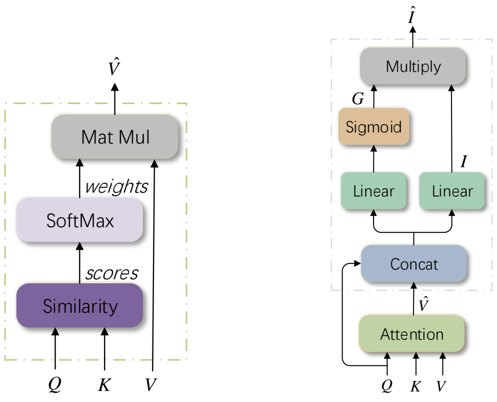
2.2. Attention Mechanisms

The attention mechanism [32, 9], which is derived from human intuition, has been widely applied and yielded significant improvements for various sequence learning tasks. It first calculates an importance score for each candidate vector, then normalizes the scores to weights using the softmax function, finally applies these weights to the candidates to generate the attention result, a weighted average vector [42]. There are other attention mechanisms such as: spatial and channel-wise attention [6], adaptive attention [27], stacked attention [46], multi-level attention [51], multi-head attention and self-attention [35].

Recently, Vaswani et al. [35] showed that solely using self-attention can achieve state-of-the-art results for machine translation. Several works extend the idea of employing self-attention to some tasks [40, 19] in computer vision, which inspires us to apply self-attention to image captioning to model relationships among objects in an image.

2.3. Other Work

AoA generates an attention gate and an information vector via two linear transformations and applies the gate to the vector to add a second attention, where the techniques are similar to some other work: GLU [10], which replaces RNN and CNN to capture long-range dependencies for language modeling; multi-modal fusion [43, 14, 22, 4, 17], which models interactions between different modalities (*e.g.* text and image) and combines information from them;



(a) Attention

(b) Attention on Attention

Figure 2: Attention and "*Attention on Attention*" (AoA). **(a)** The attention module generates some weighted average $\hat{\boldsymbol{V}}$ based on the similarity scores between $\boldsymbol{Q}$ and $\boldsymbol{K}$; **(b)** AoA generates the "information vector" $\boldsymbol{I}$ and "attention gate" $\boldsymbol{G}$, and adds another attention via element-wise multiplication.

LSTM/GRU, which uses gates and memories to model its inputs in a sequential manner.

2.4. Summarization

We summarize the differences between our method and the work discussed above, as follows: We apply *Attention on Attention* (AoA) to image captioning in this paper; AoA is a general extension to attention mechanisms and can be applied to any of them; AoA determines the relevance between the attention result and query, while multi-modal fusion combines information from different modalities; AoA requires only one "attention gate" but no hidden states. In contrast, LSTM/GRU requires hidden states and more gates, and is applicable only to sequence modeling.

3. Method

We first introduce the *Attention on Attention* (AoA) module and then show how we derive AoANet for image captioning by applying AoA to the image encoder and the caption decoder.

3.1. Attention on Attention

An attention module $f_{att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$ operates on some queries, keys and values and generates some weighted average vectors (denoted by $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$ and $\hat{\boldsymbol{V}}$ respectively), in Figure 2(a). It first measures the similarities between $\boldsymbol{Q}$ and $\boldsymbol{K}$ and then uses the similarity scores to compute weighted average vectors over $\boldsymbol{V}$, which can be formulated as:

$$a_{i,j} = f_{sim}(\boldsymbol{q}_i, \boldsymbol{k}_j), \alpha_{i,j} = \frac{e^{a_{i,j}}}{\sum_j e^{a_{i,j}}} \quad (1)$$

$$\hat{\boldsymbol{v}}_{\boldsymbol{i}} = \sum_j \alpha_{i,j} \boldsymbol{v}_j \quad (2)$$

where $\boldsymbol{q}_i \in \boldsymbol{Q}$ is the i^{th} query, $\boldsymbol{k}_j \in \boldsymbol{K}$ and $\boldsymbol{v}_j \in \boldsymbol{V}$ are the j^{th} key/value pair; f_{sim} is a function that computes the similarity score of each $\boldsymbol{k}_j$ and $\boldsymbol{q}_i$; and $\hat{\boldsymbol{v}}_i$ is the attended vector for the query $\boldsymbol{q}_i$.

The attention module outputs a weighted average for each query, no matter whether or how $\boldsymbol{Q}$ and $\boldsymbol{K}/\boldsymbol{V}$ are related. Even when there is no relevant vectors, the attention module still generates a weighted average vector, which can be irrelevant or even misleading information.

Thus we propose the AoA module (as shown in Figure 2(b)) to measure the relevance between the attention result and the query. The AoA module generates an "information vector" $\boldsymbol{i}$ and an "attention gate" $\boldsymbol{g}$ via two separate linear transformations, which are both conditioned on the attention result and the current context (*i.e.* the query) $\boldsymbol{q}$:

$$\boldsymbol{i} = W_q^i \boldsymbol{q} + W_v^i \hat{\boldsymbol{v}} + b^i \quad (3)$$

$$\boldsymbol{g} = \sigma(W_q^g \boldsymbol{q} + W_v^g \hat{\boldsymbol{v}} + b^g) \quad (4)$$

where $W_q^i, W_v^i, W_q^g, W_v^g \in \mathbb{R}^{D \times D}$, $b^i, b^g \in \mathbb{R}^D$, and D is the dimension of $\boldsymbol{q}$ and $\boldsymbol{v}$; $\hat{\boldsymbol{v}} = f_{att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$ is the attention result, f_{att} is an attention module and σ denotes the sigmoid activation function.

Then AoA adds another attention by applying the attention gate to the information vector using element-wise multiplication and obtains the attended information $\hat{\boldsymbol{i}}$:

$$\hat{\boldsymbol{i}} = \boldsymbol{g} \odot \boldsymbol{i} \quad (5)$$

where $\odot$ denotes element-wise multiplication. The throughout pipeline of AoA is formulated as:

$$\begin{aligned} \text{AoA}(f_{att}, \boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) &= \sigma(W_q^g \boldsymbol{Q} + W_v^g f_{att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) + b^g) \\ &\odot (W_q^i \boldsymbol{Q} + W_v^i f_{att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) + b^i) \quad (6) \end{aligned}$$

3.2. AoANet for Image Captioning

We build the model, AoANet, for image captioning based on the encoder/decoder framework (Figure 3), where both the encoder and the decoder are incorporated with an AoA module.

3.2.1 Encoder with AoA

For an image, we first extract a set of feature vectors $\boldsymbol{A} = \{\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_k}\}$ using a CNN or R-CNN based network,

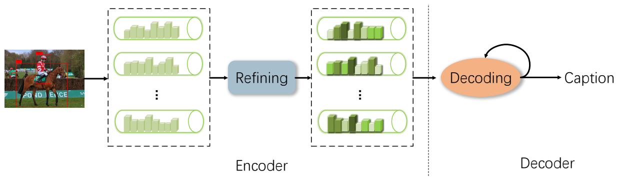


Figure 3: Overview of the encoder/decoder framework of AoANet. A refining module is added in the encoder to model relationships of objects in the image.

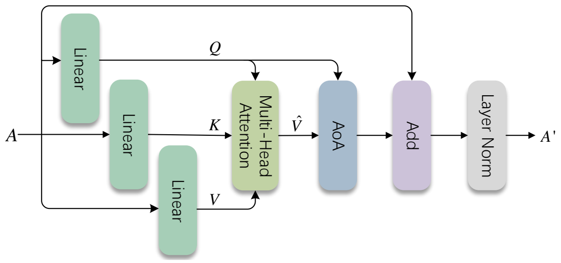


Figure 4: The refining module in the image encoder, where AoA and the self-attentive multi-head attention refine the representations of feature vectors by modeling relationships among them.

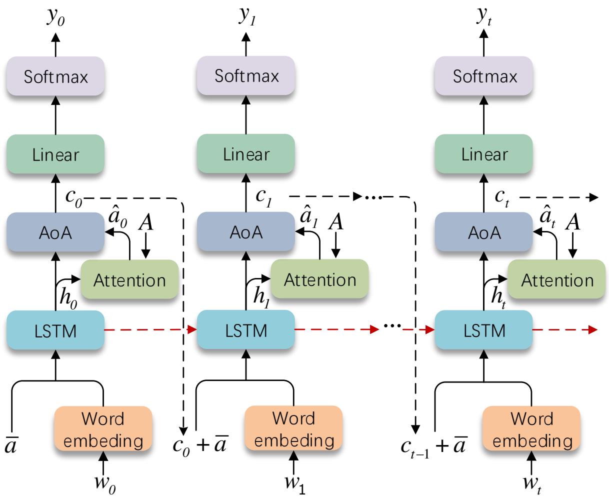


Figure 5: The caption decoder of AoANet, which contains an LSTM, an AoA module and a word prediction module.

where $\boldsymbol{a}_i \in \mathbb{R}^D$, k is the number of vectors in $\boldsymbol{A}$, and D is the dimension of each vector.

Instead of directly feeding these vectors to the decoder, we build a refining network which contains an AoA module to refine their representations (Figure 4). The AoA module in the encoder, notated as AoA^E, adopts the multi-head attention function [35] where $\boldsymbol{Q}, \boldsymbol{K}$, and $\boldsymbol{V}$ are three individual linear projections of the feature vectors $\boldsymbol{A}$. The AoA module is followed by a residual connection [18] and layer normalization [3]:

$$\begin{aligned}\boldsymbol{A}' = \text{LayerNorm}(&\boldsymbol{A}+ \\ &\text{AoA}^E(f_{mh-att}, W^{Q_e}\boldsymbol{A}, W^{K_e}\boldsymbol{A}, W^{V_e}\boldsymbol{A}))\end{aligned} \quad (7)$$

where $W^{Q_e}, W^{K_e}, W^{V_e} \in \mathbb{R}^{D\times D}$ are three linear transformation matrixes. f_{mh-att} is the multi-head attention function which divides each $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ into $H = 8$ slices along the channel dimension, and employs a scaled dot-product attention function $f_{dot-att}$ to each slice $\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i$, then concatenates the results of each slice to form the final attended vector.

$$f_{mh-att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(head_1, ..., head_H) \quad (8)$$

$$head_i = f_{dot-att}(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i) \quad (9)$$

$$f_{dot-att}(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i) = \text{softmax}(\frac{\boldsymbol{Q}_i\boldsymbol{K}_i^T}{\sqrt{d}})\boldsymbol{V}_i \quad (10)$$

In this refining module, the self-attentive multi-head attention module seeks the interactions among objects in the image, and AoA is applied to measure how well they are related. After refining, we update the feature vectors $\boldsymbol{A} \leftarrow \boldsymbol{A}'$. The refining module doesn't change the dimension of $\boldsymbol{A}$, and thus can be stacked for N times ($N = 6$ in this paper).

Note that the refining module adopts a different structure from that of the original transformer encoder [35] as the feed-forward layer is dropped, which is optional and the change is made for the following two reasons: 1) the feed-forward layer is added to provide non-linear representations, which is also realized by applying AoA; 2) dropping the feed-forward layer does not change the performances perceptually of AoANet but gives simplicity.

3.2.2 Decoder with AoA

The decoder (Figure 5) generates a sequence of caption $\boldsymbol{y}$ with the (refined) feature vectors $\boldsymbol{A}$.

We model a context vector $\boldsymbol{c}_t$ to compute the conditional probabilities on the vocabulary:

$$p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, I) = \text{softmax}(W_p\boldsymbol{c}_t) \quad (11)$$

where $W_p \in \mathbb{R}^{D\times|\Sigma|}$ is the weight parameters to be learnt and $|\Sigma|$ the size of the vocabulary.

The context vector $\boldsymbol{c}_t$ saves the decoding state and the newly acquired information, which is generated with the attended feature vector $\hat{\boldsymbol{a}}_t$ and the output $\boldsymbol{h}_t$ of an LSTM,

Table 1: Performance of our model and other state-of-the-art methods on MS-COCO "Karpathy" test split, where B@N, M, R, C and S are short for BLEU@N, METEOR, ROUGE-L, CIDEr-D and SPICE scores. All values are reported as percentage (%). Σ indicates an ensemble or fusion.

Model	**Cross-Entropy Loss**						**CIDEr-D Score Optimization**					
Metric	B@1	B@4	M	R	C	S	B@1	B@4	M	R	C	S
	Single Model											
LSTM [37]	-	29.6	25.2	52.6	94.0	-	-	31.9	25.5	54.3	106.3	-
SCST [31]	-	30.0	25.9	53.4	99.4	-	-	34.2	26.7	55.7	114.0	-
LSTM-A [50]	75.4	35.2	26.9	55.8	108.8	20.0	78.6	35.5	27.3	56.8	118.3	20.8
Up-Down [2]	77.2	36.2	27.0	56.4	113.5	20.3	79.8	36.3	27.7	56.9	120.1	21.4
RFNet [20]	76.4	35.8	27.4	56.8	112.5	20.5	79.1	36.5	27.7	57.3	121.9	21.2
GCN-LSTM [49]	77.3	36.8	27.9	57.0	116.3	20.9	80.5	38.2	28.5	58.3	127.6	22.0
SGAE [44]	-	-	-	-	-	-	**80.8**	38.4	28.4	58.6	127.8	22.1
AoANet (Ours)	**77.4**	**37.2**	**28.4**	**57.5**	**119.8**	**21.3**	80.2	**38.9**	**29.2**	**58.8**	**129.8**	**22.4**
	Ensemble/Fusion											
SCST [31]$^{\Sigma}$	-	32.8	26.7	55.1	106.5	-	-	35.4	27.1	56.6	117.5	-
RFNet [20]$^{\Sigma}$	77.4	37.0	27.9	57.3	116.3	20.8	80.4	37.9	28.3	58.3	125.7	21.7
GCN-LSTM [49]$^{\Sigma}$	77.4	37.1	28.1	57.2	117.1	21.1	80.9	38.3	28.6	58.5	128.7	22.1
SGAE [44]$^{\Sigma}$	-	-	-	-	-	-	81.0	39.0	28.4	58.9	129.1	22.2
AoANet (Ours)$^{\Sigma}$	**78.7**	**38.1**	**28.5**	**58.2**	**122.7**	**21.7**	**81.6**	**40.2**	**29.3**	**59.4**	**132.0**	**22.8**

Table 2: Leaderboard of various methods on the online MS-COCO test server.

Model	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr-D	
Metric	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST [31]	78.1	93.7	61.9	86.0	47.0	75.9	35.2	64.5	27.0	35.5	56.3	70.7	114.7	116.0
LSTM-A [50]	78.7	93.7	62.7	86.7	47.6	76.5	35.6	65.2	27.0	35.4	56.4	70.5	116.0	118.0
Up-Down [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 [20]	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 [49]	-	-	65.5	89.3	50.8	80.3	38.7	69.7	28.5	37.6	58.5	73.4	125.3	126.5
SGAE [44]	**81.0**	**95.3**	65.6	89.5	50.7	80.4	38.5	69.7	28.2	37.2	58.6	73.6	123.8	126.5
AoANet (Ours)	**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**

where $\hat{\boldsymbol{a}}_t$ is the attended result from an attention module which could have a single head or multiple heads.

The LSTM in the decoder models the caption decoding process. Its input consists of the embedding of the input word at current time step, and a visual vector $(\bar{\boldsymbol{a}} + \boldsymbol{c}_{t-1})$, where $\bar{\boldsymbol{a}} = \frac{1}{k}\sum_i \boldsymbol{a}_i$ denotes the mean pooling of $\boldsymbol{A}$ and $\boldsymbol{c}_{t-1}$ denotes the context vector at previous time step ($\boldsymbol{c}_{-1}$ is initialized to zeros at the beginning step):

$$\boldsymbol{x}_t = [W_e \Pi_t, \bar{\boldsymbol{a}} + \boldsymbol{c}_{t-1}] \tag{12}$$

$$\boldsymbol{h}_t, \boldsymbol{m}_t = \text{LSTM}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}, \boldsymbol{m}_{t-1}) \tag{13}$$

where $W_e \in \mathbb{R}^{E\times|\Sigma|}$ is a word embedding matrix for a vocabulary Σ, and Π_t is one-hot encoding of the input word w_t at time step t.

As shown in Figure 5, for the AoA decoder, $\boldsymbol{c}_t$ is obtained from an AoA module, notated as AoA^D:

$$\boldsymbol{c}_t = \text{AoA}^D(f_{mh-att}, W^{Q_d}[\boldsymbol{h}_t], W^{K_d}A, W^{V_d}A) \tag{14}$$

where $W^{Q_e}, W^{K_e}, W^{V_e} \in \mathbb{R}^{D\times D}$; $\boldsymbol{h}_t, \boldsymbol{m}_t \in \mathbb{R}^D$ the hidden states of the LSTM and $\boldsymbol{h}_t$ serves as the attention query.

3.3. Training and Objectives

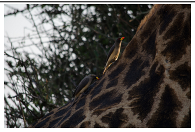
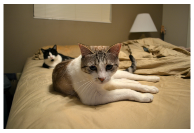
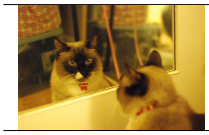
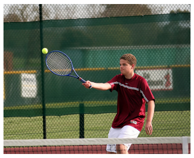
Training with Cross Entropy Loss. We first train AoANet by optimizing the cross entropy (XE) loss L_{XE}:

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log(p_\theta(\boldsymbol{y}_t^* \mid \boldsymbol{y}_{1:t-1}^*)) \tag{15}$$

where $\boldsymbol{y}_{1:T}^*$ denotes the target ground truth sequence.

CIDEr-D Score Optimization. Then we directly optimize the non-differentiable metrics with Self-Critical Sequence

Table 3: Examples of captions generated by AoANet and a baseline model as well as the corresponding ground truths.

Image	Captions
	**AoANet**: Two birds sitting on top of a giraffe. **Baseline**: A bird sitting on top of a tree. **GT1**. Two birds going up the back of a giraffe. **GT2**. A large giraffe that is walking by some trees. **GT3**. Two birds are sitting on a wall near the bushes.
	**AoANet**: Two cats laying on top of a bed. **Baseline**: A black and white cat laying on top of a bed. **GT1**. A couple of cats laying on top of a bed. **GT2**. Two cats laying on a big bed and looking at the camera. **GT3**. A couple of cats on a mattress laying down.
	**AoANet**: A cat looking at its reflection in a mirror. **Baseline**: A cat is looking out of a window. **GT1**. A cat looking at his reflection in the mirror. **GT2**. A cat that is looking in a mirror. **GT3**. A cat looking at itself in a mirror.
	**AoANet**: A young boy hitting a tennis ball with a tennis racket. **Baseline**: A young man holding a tennis ball on a court. **GT1**. A guy in a maroon shirt is holding a tennis racket out to hit a tennis ball. **GT2**. A man on a tennis court that has a racquet. **GT3**. A boy hitting a tennis ball on the tennis court.

Training [31] (SCST):

$$L_{RL}(\theta) = -\mathbf{E}_{\boldsymbol{y}_{1:T}\sim p_\theta}[r(\boldsymbol{y}_{1:T})] \tag{16}$$

where the reward $r(\cdot)$ uses the score of some metric (*e.g.* CIDEr-D [36]). The gradients can be approximated:

$$\nabla_\theta L_{RL}(\theta) \approx -(r(\boldsymbol{y}^s{}_{1:T}) - r(\hat{\boldsymbol{y}}_{1:T}))\nabla_\theta \log p_\theta(\boldsymbol{y}^s_{1:T}) \tag{17}$$

$\boldsymbol{y}^s$ means it's a result sampled from probability distribution, while $\hat{\boldsymbol{y}}$ indicates a result of greedy decoding.

3.4. Implementation Details

We employ a pre-trained Faster-RCNN [30] model on ImageNet [11] and Visual Genome [24] to extract bottom-up feature vectors of images [2]. The dimension of the original vectors is 2048 and we project them to a new space with the dimension of $D = 1024$, which is also the hidden size of the LSTM in the decoder. As for the training process, we train AoANet under XE loss for 30 epochs with a mini batch size of 10, and ADAM [23] optimizer is used with a learning rate initialized by 2e-4 and annealed by 0.8 every 3 epochs. We increase the scheduled sampling probability by 0.05 every 5 epochs [5]. We optimize the CIDEr-D score with SCST for another 15 epochs with an initial learning rate of 2e-5 and annealed by 0.5 when the score on the validation split does not improve for some training steps.

4. Experiments

4.1. Dataset

We evaluate our proposed method on the popular MS COCO dataset [26]. MS COCO dataset contains 123,287 images labeled with 5 captions for each, including 82,783 training images and 40,504 validation images. MS COCO provides 40,775 images as test set for online evaluation as well. The offline "Karpathy" data split [21] is used for the offline performance comparisons, where 5,000 images are used for validation, 5,000 images for testing and the rest for training. We convert all sentences to lower case, and drop the words that occur less than 5 times and end up with a vocabulary of 10,369 words. We use different metrics, including BLEU [29], METEOR [33], ROUGE-L [13], CIDEr-D [36] and SPICE [1], to evaluate the proposed method and compare with other methods. All the metrics are computed with the publicly released code[1].

4.2. Quantitative Analysis

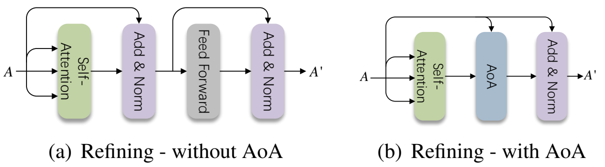
Offline Evaluation. We report the performance on the offline test split of our model as well as the compared models in Table 1. The models include: LSTM [37], which encodes the image using CNN and decodes it using LSTM; SCST [31], which employs a modified visual attention and is the first to use SCST to directly optimize the evaluation metrics; Up-Down [2], which employs a two-LSTM layer model with bottom-up features extracted from Faster-RCNN; RFNet [20], which fuses encoded features from multiple CNN networks; GCN-LSTM [49], which predicts visual relationships between every two entities in the image and encodes the relationship information into feature vectors; and SGAE [44], which introduces auto-encoding scene graphs into its model.

For fair comparison, all the models are first trained under XE loss and then optimized for CIDEr-D score. For the XE loss training stage in Table 1, it can be seen that our single model achieves the highest scores among all compared methods in terms of all metrics even comparing with the ensemble of their models. As for the CIDEr-D score optimization stage, an ensemble of 4 models with different parameter initialization of AoANet outperforms all other models and sets a new state-of-the-art performance of 132.0 CIDEr-D score.

Online Evaluation. We also evaluate our model on the online COCO test server[2] in Table 2. The results of AoANet are evaluated by an ensemble of 4 models trained on the "Karpathy" training split. AoANet achieves the highest scores for most metrics except a slightly lower one for BLEU-1 (C40).

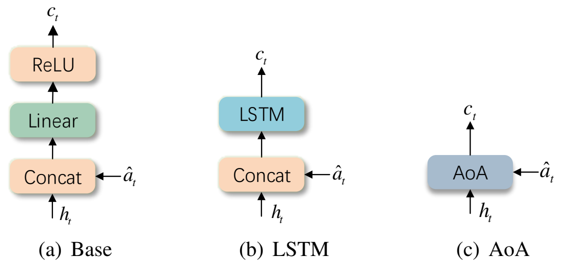
[1] https://github.com/tylin/coco-caption

[2] https://competitions.codalab.org/competitions/3221#results



(a) Refining - without AoA (b) Refining - with AoA

Figure 6: Refining modules w/o and w/ AoA.



(a) Base (b) LSTM (c) AoA

Figure 7: Different schemes for decoders to model $\boldsymbol{c}_t$.

4.3. Qualitative Analysis

Table 3 shows a few examples with images and captions generated by our AoANet and a strong baseline as well as the human-annotated ground truths. We derive the baseline model by re-implementing the Up-Down [2] model with the settings of AoANet. From these examples, we find that the baseline model generates captions which are in line with the logic of language but inaccurate for the image content, while AoANet generates accurate captions in high quality. More specifically, our AoANet is superior in the following two aspects: 1) AoANet counts objects of the same kind more accurately. There are two birds/cats in the image of the first/second example. However, the baseline model finds only one while our AoANet counts correctly; 2) AoANet figures out the interactions of objects in an image. For example, AoANet knows that the birds are on top of a *giraffe* but not the *tree*, in the first example; the boy is *hitting* the tennis ball with a racket but not *holding*, in the fourth example. AoANet has these advantages because it can figure out the connections among objects and also knows how they are connected: in the encoder, the refining module uses self-attention to seek interactions among objects and uses AoA to measure how well they are related; in the decoder, AoA helps to filter out irrelative objects which don't have the required interactions and only keeps the related ones. While the baseline model generates captions which are logically right but might not match the image contents.

4.4. Ablative Analysis

To quantify the impact of the proposed AoA module, we compare AoANet against a set of other ablated models with various settings. We first design the "base" model which doesn't have a refining module in its encoder and adopts a "base" decoder in Figure 7(a), using a linear transformation to generate the context vector $\boldsymbol{c}_t$.

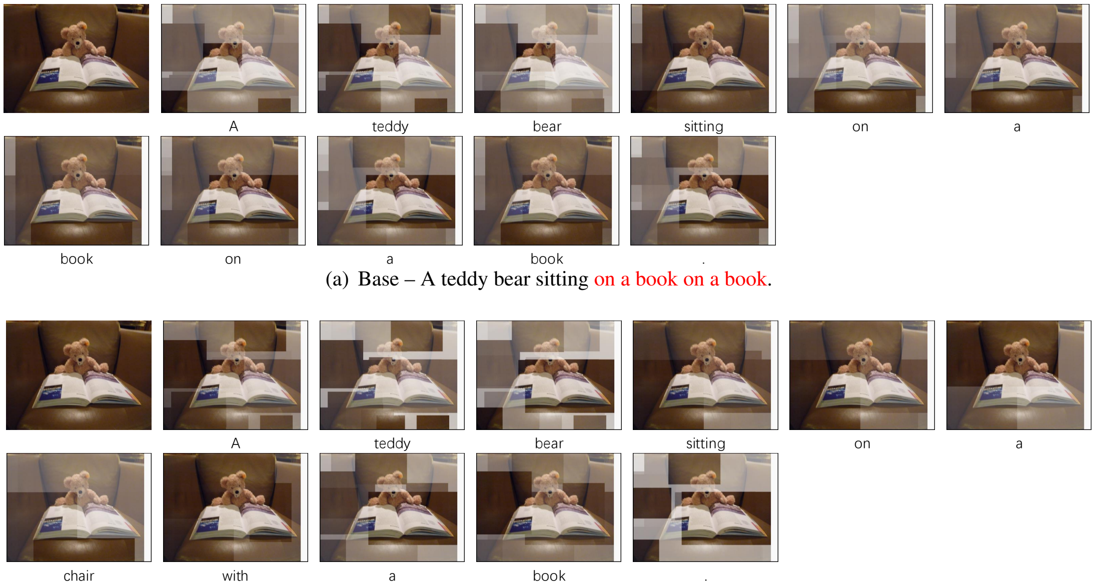
Table 4: Settings and results of ablation studies. The results are reported after XE training stage.

Model	B@1	B@4	R	C
Base	75.7	34.9	56.0	109.5
+ Enc: Refine (w/o AoA)	**77.0**	35.6	56.4	112.5
+ Enc: Refine (w/ AoA)	76.7	**36.1**	**56.7**	**114.5**
+ Dec: LSTM	76.8	35.9	56.6	113.5
+ Dec: AoA	76.6	35.8	56.6	113.8
+ Dec: LSTM + AoA	*unstable training process*			
+ Dec: MH-Att	75.8	34.8	56.0	109.6
+ Dec: MH-Att, LSTM	76.6	35.8	**56.7**	113.8
+ Dec: MH-Att, AoA	**76.9**	**36.1**	56.6	**114.3**
Full: AoANet	**77.4**	**37.2**	**57.5**	**119.8**

Effect of AoA on the encoder. To evaluate the effect of applying AoA to the encoder, we design a refining module without AoA, which contains a self-attention module and a following feed-forward transition, in Figure 6(a). From Table 4 we observe that refining the feature representations brings positive effects, and adding a refining module without AoA improves the CIDEr-D score of "base" by 3.0. We then apply AoA to the attention mechanism in the refining module and we drop the feed-forward layer. The results show that our AoA further improves the CIDEr-D score by 2.0.

Effect of AoA on the decoder. We compare the performance of using different schemes to model the context vector $\boldsymbol{c}_t$: "base" (Figure 7(a)), via a linear transformation; "LSTM" (Figure 7(b)), via an LSTM; AoA (Figure 7(c)), by applying AoA. We conduct experiments with both single attention and multi-head attention (MH-Att). From Table 4, we observe that replacing single attention with multi-head attention brings slightly better performances. Using LSTM improves the performance of the base model, and AoA further outperforms LSTM. Comparing to LSTM or GRU, which uses some memories (hidden states) and gates to model attention results in a sequential manner, AoA is more light-weighted as it involves only two linear transformation and requires little computation. Even so, AoA still outperforms LSTM. We also find that the training process of "LSTM + AoA" (building AoA upon LSTM) is unstable and could reach a sub-optimal point, which indicates that stacking more gates doesn't provide further performance improvements.

To qualitatively show the effect of AoA, we visualize the caption generation process in Figure 8 with attended image regions for each decoding time step. Two models are compared: the "base" model, which doesn't incorporate the



(a) Base – A teddy bear sitting on a book on a book.

(b) AoA – A teddy bear sitting on a chair with a book.

Figure 8: Visualization of attention regions in the caption generation process for the “base” model and “decoder with AoA”. The “base” model can be easily misled by irrelevant attention while “decoder with AoA” is less likely so.

AoA module, and “decoder with AoA”, which employs an AoA module in its caption decoder. Observing the attended image regions in Figure 8, we find that the attention module isn’t always reliable for the caption decoder to generate a word, and directly using the attention result might result in wrong captions. In the example, the book is attended by the base model when generating the caption fragment *“A teddy bear sitting on a ...”*. As a result, the base model outputs *“book”* for the next word, which is not consistent to what the image shows since the teddy bear is actually sitting on a *chair* but not on a *book*. In contrast, “decoder with AoA” is less likely to be misled by irrelevant attention results, because the AoA module in it adds another attention on the attention result, which suppress the irrelevant/misleading information and keeps only the useful.

4.5. Human Evaluation

We follow the practice in [44] and invited 30 evaluators to evaluate 100 randomly selected images. For each image, we show the evaluators two captions generated by “decoder with AoA” and the “base” model in random order, and ask them which one is more descriptive. The percentages of “decoder with AoA”, “base”, and *comparative* are 49.15%, 21.2%, and 29.65% respectively, which shows the effectiveness of AoA as confirmed by the evaluators.

4.6. Generalization

To show the general applicability of AoA, we perform experiments on a video captioning dataset, MSR-VTT [41]: we use ResNet-101 [18] to extract feature vectors from sampled 20 frames of each video and then pass them to a bi-LSTM and a decoder, “base” or “decoder with AoA”. We find that “decoder with AoA” improves “base” from BLEU-4: 33.53,CIDEr-D: 38.83, ROUGE-L 56.90 to 37.22, 42.44, 58.32, respectively, which shows that AoA is also promising for other tasks which involve attention mechanisms.

5. Conclusion

In this paper, we propose the *Attention on Attention* (AoA) module, an extension to conventional attention mechanisms, to address the irrelevant attention issue. Furthermore, we propose AoANet for image captioning by applying AoA to both the encoder and decoder. More remarkably, we achieve a new state-of-the-art performance with AoANet. Extensive experiments conducted on the MS COCO dataset demonstrate the superiority and general applicability of our proposed AoA module and AoANet.

Acknowledgment

This project was supported by Shenzhen Key Laboratory for Intelligent Multimedia and Virtual Reality (ZDSYS201703031405467), National Natural Science Foundation of China (NSFC, No.U1613209, 61872256, 61972217), and National Engineering Laboratory for Video Technology - Shenzhen Division. We would also like to thank Qian Wu, Yaxian Xia and Qixiang Ye, as well as the anonymous reviewers for their insightful comments.

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 visual question answering. In *CVPR*, 2018.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Hedi Ben-younes, Remi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *ICCV*, 2017.

[5] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NeurIPS*, 2015.

[6] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, 2017.

[7] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. In *ECCV*, 2018.

[8] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[9] Maurizio Corbetta and Gordon L Shulman. Control of goal-directed and stimulus-driven attention in the brain. *Nature Reviews Neuroscience*, 3(3):201–215, 2002.

[10] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *ICLR*, 2016.

[11] Jia Deng, Wenjie Dong, Richard Socher, Lijia Li, Kehui Li, and Li Feifei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[12] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015.

[13] Carlos Flick. Rouge: A package for automatic evaluation of summaries. In *The Workshop on Text Summarization Branches Out*, 2004.

[14] Akira Fukui, Huk Park Dong, Daylen Yang, Anna Rohrbach, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016.

[15] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. Stylenet: Generating attractive visual captions with styles. In *CVPR*, July 2017.

[16] 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.

[17] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven C. H. Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra- and inter-modality attention flow for visual question answering. In *CVPR*, June 2019.

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

[19] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, 2018.

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

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

[22] Jin Hwa Kim, Kyoung Woon On, Jeonghee Kim, Jung Woo Ha, and Byoung Tak Zhang. Hadamard product for low-rank bilinear pooling. In *ICLR*, 2017.

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

[24] 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*, 2017.

[25] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013.

[26] 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.

[27] 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.

[28] Margaret Mitchell, Xufeng Han, Jesse Dodge, Alyssa Mensch, Amit Goyal, Alex Berg, Kota Yamaguchi, Tamara Berg, Karl Stratos, and Hal Daumé III. Midge: Generating image descriptions from computer vision detections. In *European Chapter of the Association for Computational Linguistics*, 2012.

[29] Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[30] Shaoqing Ren, Kaiming He, Ross B Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

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

[32] Ronald A Rensink. The dynamic representation of scenes. *Visual Cognition*, 7:17–42, 2000.

[33] Banerjee Satanjeev. Meteor : An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL*, 2005.

[34] Richard Socher and Li Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *CVPR*, 2010.

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

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

[37] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2015.

[38] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-mil: Continuation multiple instance learning for weakly supervised object detection. In *CVPR*, 2019.

[39] Fang Wan, Pengxu Wei, Zhenjun Han, Jianbin Jiao, and Qixiang Ye. Min-entropy latent model for weakly supervised object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[40] Xiaolong Wang, Ross B Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[41] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[42] 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.

[43] Gao Yang, Oscar Beijbom, Zhang Ning, and Trevor Darrell. Compact bilinear pooling. In *CVPR*, 2015.

[44] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, June 2019.

[45] Yezhou Yang, Ching Lik Teo, Hal Daumé III, and Yiannis Aloimonos. Corpus-guided sentence generation of natural images. In *EMNLP*, 2011.

[46] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *CVPR*, June 2016.

[47] Zhilin Yang, Ye Yuan, Yuexin Wu, Ruslan Salakhutdinov, and William W. Cohen. Review networks for caption generation. In *NeurIPS*, 2016.

[48] Benjamin Z Yao, Xiong Yang, Liang Lin, Mun Wai Lee, and Song-Chun Zhu. I2t: Image parsing to text description. *Proceedings of the IEEE*, 98(8):1485–1508, 2010.

[49] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *ECCV*, 2018.

[50] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. In *ICCV*, 2017.

[51] Dongfei Yu, Jianlong Fu, Tao Mei, and Yong Rui. Multi-level attention networks for visual question answering. In *CVPR*, July 2017.