The Summary of "Attention on Attention for Image Captioning" [1] Paper

Furkan Gül

I. Motivation

Recently, attention mechanisms enable advances in the neural machine translation era. It also becomes widely used in current encoder & decoder models for image captioning problems which results in impressive achievements. In these traditional frameworks, a CNN (encoder framework) encodes an input image to a set of feature vectors. Then, an RNN (decoder framework) decodes this set of feature vectors to words. This decoding process is lead by the attention mechanism which outputs a weighted average for the whole set of feature vectors at each time step. There are two possible reasons why decoder part is misled and

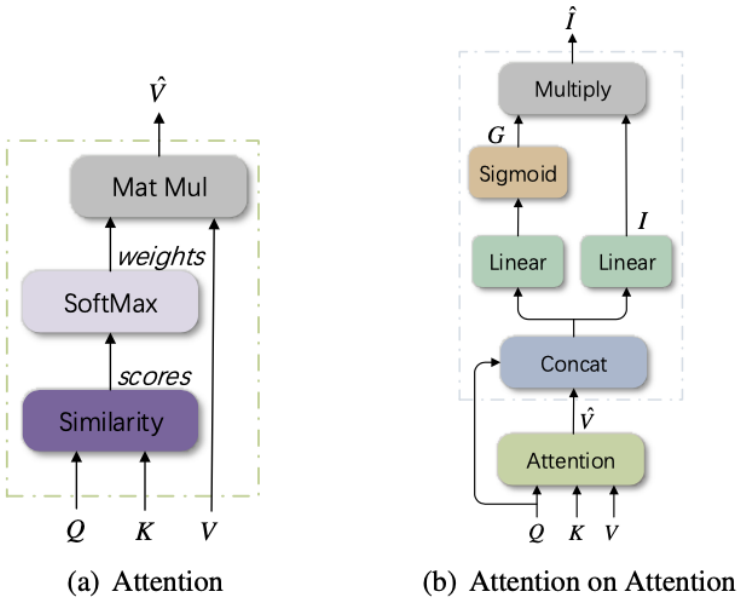

Fig. 1. (a) The attention mechanism creates some weighted average $\hat{\boldsymbol{V}}$ based on the similarity scores between $\boldsymbol{Q}$ and $\boldsymbol{K}$ (b) AoA creates the "information vector", $\boldsymbol{I}$, and "attention gate", $\boldsymbol{G}$, and adds a new attention mechanism via element-wise multiplication [1].

gives deceptive results. The first reason arises from the poor performance of the attention part. The second reason is that there could be no significant information from the feature vectors at all. The first reason is more natural and hard to be avoided. For the second case, even a specific query requirement is not satisfied, the attention mechanism still returns a weighted average vector over the feature vectors which would be completely irrelevant to the context. To solve this issue, Attention on Attention (AoA) is proposed by this paper. AoA network simply adds another attention mechanism to conventional attention.

II. Approach

A. Attention on Attention

As it can be seen in Figure 1(a), a general attention network, $f_{att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})$, processes over queries (denoted by $\boldsymbol{Q}$), keys (denoted by $\boldsymbol{K}$), and values (denoted by $\boldsymbol{V}$) to output weighted average vectors (denoted by $\hat{\boldsymbol{V}}$). Even if there is no relation between $\boldsymbol{Q}$ and $\boldsymbol{K}/\boldsymbol{V}$, then this traditional attention network gives an output of weighted average for each query. Eventually, this irrelevant output information will mislead the decoder part. As it is shown in Figure 1(b), AoA network tries to evaluate and assess the relationship between the attention result and the query (the current context). The formulation for AoA network can be seen in Equation 1:

$$AoA(f_{att}, \boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \sigma(W_q^g \boldsymbol{Q} + W_v^g f_{att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) + b^g) \odot (W_q^i \boldsymbol{Q} + W_v^i f_{att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) + b^i) \tag{1}$$

where i denotes information, g denotes attention gate, $\odot$ represents element-wise multiplication, $W_q^i, W_v^i, W_q^g, W_v^g \in \mathbb{R}^{DxD}$, $b^i, b^g \in \mathbb{R}^D$, D is the dimension of the query and the value, σ denotes the sigmoid activation function.

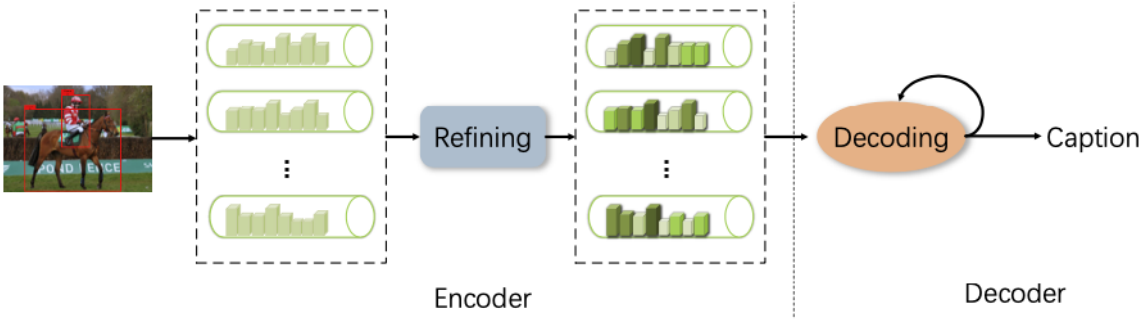

Fig. 2. Encoder & decoder framework of AoANet. A refining module is added in the encoder to model relationships of objects in the image [1].

B. AoANet for Captioning

AoANet model is built on encoder & decoder framework by adding an AoA mechanism mentioned in the previous section to both encoder and decoder. AoANet for image captioning can be seen in Figure 2.

C. Encoder with AoA Mechanism

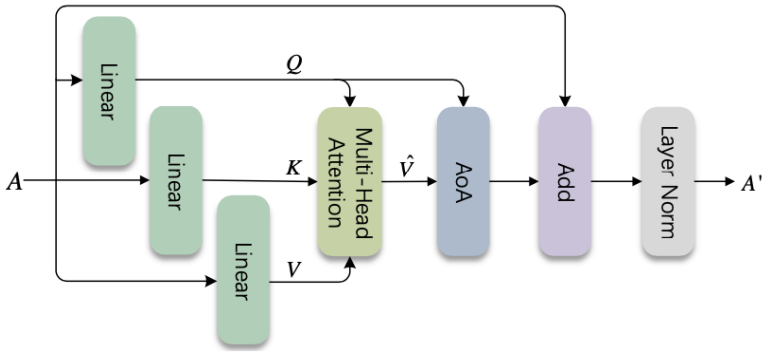

Fig. 3. The refining network in the encoder part, where AoA and the self-attentive multi-head attention refine the representations of feature vectors by modeling relationships among them [1].

A CNN or R-CNN type of model is used to extract a set of feature vectors, $\boldsymbol{A} = \boldsymbol{a_1}, \boldsymbol{a_2}, ..., \boldsymbol{a_k}$, for an input image. $\boldsymbol{a_i} \in \mathbb{R}^D$, k is the total number of vectors in $\boldsymbol{A}$, and D represents the dimension of each feature vector. As it is also seen in Figure 2, these feature vectors are not directly fed to the decoder, they are processed by a refining network which includes an AoA mechanism to improve their representations (see Figure 3). The resulting representation, $\boldsymbol{A}^{'}$, can be formulated as:

$$\boldsymbol{A}^{'} = LayerNorm(\boldsymbol{A} + AoA^E(f_{mh-att}, W^{Q_e}\boldsymbol{A}, W^{K_e}\boldsymbol{A}, W^{V_e}\boldsymbol{A})) \tag{2}$$

where AoA^E represents the AoA module in the image encoder part, $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are individual linear projections of the feature vectors, $\boldsymbol{A}$, LayerNorm means layer normalization, $W^{Q_e}, W^{K_e}, W^{V_e} \in \mathbb{R}^{DxD}$ are linear transformation matrices, f_{mh-att} denotes the multi-headed attention function with $H = 8$. Detailed formulations for the multi-headed attention can be found below:

$$f_{mh-att}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Concat(head_1, ..., head_H) \tag{3}$$

$$head_i = f_{dot-att}(\boldsymbol{Q_i}, \boldsymbol{K_i}, \boldsymbol{V_i}) \tag{4}$$

$$f_{dot-att}(\boldsymbol{Q_i}, \boldsymbol{K_i}, \boldsymbol{V_i}) = softmax(\frac{\boldsymbol{Q_i}\boldsymbol{K_i^T}}{\sqrt{d}})\boldsymbol{V_i} \tag{5}$$

where $\boldsymbol{Q_i}, \boldsymbol{K_i}, \boldsymbol{V_i}$ are one slice of the multi-headed (8) attention, $f_{dot-att}$ means a scaled dot-product attention function for each slice.

In this refining network, the multi-headed self-attention part looks for the interactions/relations among objects in an input image. Then, AoA is implemented to assess how well these interactions are related. Since there is no feed-forward layer in the

refining module, it is different than the original transformer encoder [2]. It is observed that dropping the feed-forward layer does not affect the performance of AoANet since the non-linearity that comes from the feed-forward layer is already satisfied by implementing AoA module.

D. Decoder with AoA Mechanism

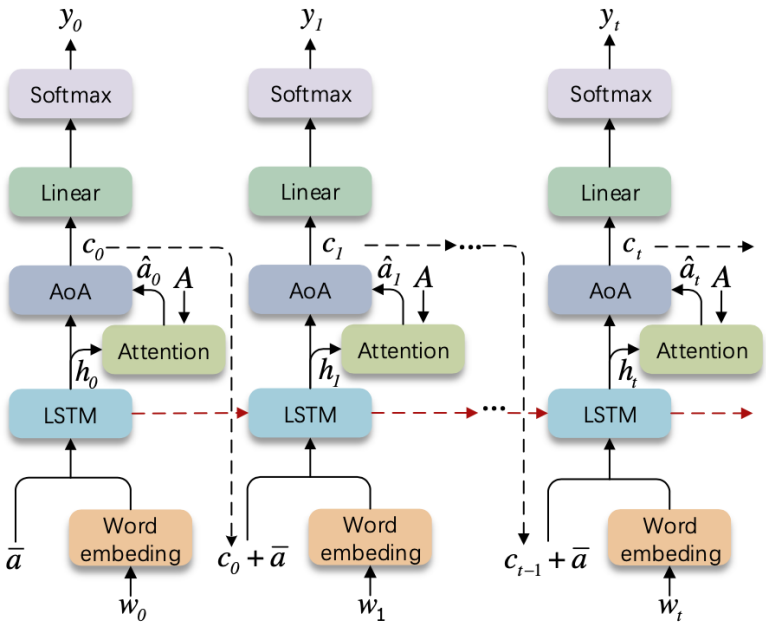

Fig. 4. Caption decoder of AoANet [1].

Decoder part seen in Figure 4 outputs a sequence of caption $\boldsymbol{y}$ together with the resulting feature vectors from the refining module, $\boldsymbol{A}$. The context vector, $\boldsymbol{c_t}$, gets the decoding state of the LSTM ($\boldsymbol{h_t}$) and the new information about interactions ($\hat{\boldsymbol{a}}_{\boldsymbol{t}}$). Here, $\hat{\boldsymbol{a}}_{\boldsymbol{t}}$ represents the attended representation coming from an attention module. Formulations for the input vector and the output hidden states of the LSTM in decoder can be seen in below two Equations.

$$\boldsymbol{x_t} = [W_e \prod_t, \bar{\boldsymbol{a}} + \boldsymbol{c_{t-1}}] \tag{6}$$

$$\boldsymbol{h_t}, \boldsymbol{m_t} = LSTM(\boldsymbol{x_t}, \boldsymbol{h_{t-1}}, \boldsymbol{m_{t-1}}) \tag{7}$$

As it is seen in Figure 4, $\boldsymbol{c_t}$ is formed by an AoA module (represented by AoA^D) as in Equation 8:

$$\boldsymbol{c_t} = AoA^D(f_{mh-att}, W^{Q_d}[\boldsymbol{h_t}], W^{K_d}A, W^{V_d}A) \tag{8}$$

where $\bar{\boldsymbol{a}} = \frac{1}{k}\sum_i \boldsymbol{a_i}$ represents the mean pooling of $\boldsymbol{A}$, $\boldsymbol{c_{t-1}}$ stands for the context vector at previous time step, $W_e \in \mathbb{R}^{E \times |\sum|}$ is a word embedding matrix, $\sum$ is the vocabulary size, $\prod_t$ represents the one-hot encoding of an input word (w_t) at time step t, $W^{Q_e}, W^{K_e}, W^{V_e} \in \mathbb{R}^{DxD}$ are linear transformation matrices, $\boldsymbol{h_t}, \boldsymbol{m_t} \in \mathbb{R}^D$ are the hidden states of the LSTM where $\boldsymbol{h_t}$ is used as the query for the attention.

E. Details of the Implementation

Firstly, the cross-entropy loss is used for the training of AoANet. Then, Self-Critical Sequence Sequence Training (SCST) [3] is used for the optimization of non-differentiable metrics like CIDEr-D Score.

A pre-trained Faster-RCNN [4] model on both Visual Genome [5] and ImageNet [6] datasets is applied to extract bottom-up feature vectors of input images. The original dimension of the feature vectors is 2048. But, they are projected to a new space with the dimension of $D = 1024$ in the paper. This $D = 1024$ is also the dimension of the hidden states of the LSTM in decoder part. AoANet is trained with a mini-batch size of 10 and ADAM [7] optimizer for 30 epochs. The learning rate is started with 2e-4 and is reduced by 0.8 multiplication constant for every 3 epochs. For the optimization of the CIDEr-D score, SCST is applied for another 15 epochs. Here, the learning rate starts with 2e-5 and is reduced by 0.5 multiplication constant where there is no improvement in the validation set.

III. Dataset & Results

MSCOCO [8] dataset is used in their experiments to evaluate the suggested captioning model. There are at least 5 sentence descriptions for every image in this dataset. They use Karparthy split [9] which is extensively preferred for reporting results in previous works. For both the validation and test phases with MSCOCO dataset, 5000 images are used. They convert all caption sentences to lower case, tokenize on white space, and discard words that have less than five occurrences. Thus, the vocabulary size for their experiments is fixed and 10369. For the evaluation of the caption model, following metrics are used: CIDEr [10], METEOR [11], SPICE [12], ROUGE-L [13], and BLEU [14]. All these metrics are calculated with the standard public code [1].

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**

Fig. 5. Single-model image captioning performance on the MSCOCO 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 (%).

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**

Fig. 6. Leaderboard of several models on the online MSCOCO test server.

In Figure 5, they reports their results together with LSTM [15], the SCST [3], LSTM-A [16], Up-Down [17], RFNet [18], GCN-LSTM [19], and SGAE [20] on the Karpathy test split. As it can be seen in Figure 5, AoANet model outperforms all SOTA models and becomes the new SOTA single-model for Karpathy split in terms of almost all evaluation metrics except BLUE-1 for CIDEr Score Optimization.

In Figure 6, they report the performance of their ensembled models (represented with $\sum$ sign) trained on the official MSCOCO evaluation server. At the time of submission, around the latest 2019, they take the lead in the leader board by outperforming all other submissions on the test server in terms of almost all metrics.

References

[1] Lun Huang, Wenmin Wang, J. Chen, and Xiao-Yong Wei. Attention on attention for image captioning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4633–4642, 2019.

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

[3] Steven J. Rennie, E. Marcheret, Youssef Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, 2017.

[4] Shaoqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[5] R. Krishna, Yuke Zhu, O. Groth, J. Johnson, Kenji Hata, J. Kravitz, Stephanie Chen, Yannis Kalantidis, L. Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016.

[6] Jia Deng, W. Dong, R. Socher, L. Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR 2009*, 2009.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

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

[8] Tsung-Yi Lin, M. Maire, Serge J. Belongie, J. Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. *ArXiv*, abs/1405.0312, 2014.

[9] A. Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:664–676, 2017.

[10] Ramakrishna Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2015.

[11] Michael J. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *WMT@ACL*, 2014.

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

[13] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL 2004*, 2004.

[14] Kishore Papineni, S. Roukos, T. Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[15] Oriol Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.

[16] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and T. Mei. Boosting image captioning with attributes. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4904–4912, 2017.

[17] Peter Anderson, X. He, C. Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

[18] W. Jiang, Lin Ma, Yu-Gang Jiang, W. Liu, and T. Zhang. Recurrent fusion network for image captioning. *ArXiv*, abs/1807.09986, 2018.

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

[20] X. Yang, Kaihua Tang, Hanwang Zhang, and J. Cai. Auto-encoding scene graphs for image captioning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10677–10686, 2019.