Supplementary Material for "Learning to Collocate Neural Modules for Image Captioning"

Xu Yang[1], Hanwang Zhang[1], Jianfei Cai[1,2]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore,
[2]Faculty of Information Technology, Monash University, Australia,
s170018@e.ntu.edu.sg,{hanwangzhang@,ASJFCai@}ntu.edu.sg

This supplementary document will further detail the following aspects in the main paper: A. Network Architecture, B. Details of Human Evaluations, C. More Qualitative Examples.

A. Network Architecture

Here, we introduce the detailed network architectures of all the components in our model, which includes four neural modules, a module controller, and decoders.

A.1. Neural Modules

In Section 3.1 of the main paper, we show how to use four neural modules to generate the orthogonal knowledge from the image. The detail structures of these four modules are respectively listed in the following tables: 1) OBJECT module in Table A, 2) ATTRIBUTE module in Table B, 3) RELATION module in Table C, and 4) FUNCTION module in Table D. In particular, the input vector $\boldsymbol{c}$ of FUNCTION module in Table D (1) is the output of an LSTM in the language decoder, and we will specify this context vector in Section A.3.

A.2. Module Controller

In Section 3.2.1 of the main paper, we discuss how to use module controller to softly fuse four vectors generated by attention networks and FUNCTION module. The common structure of three attention networks used in Eq.(7) and the detail process of soft fusion in Eq.(8) are demonstrated in Table E and F, respectively. Specifically, the hidden vector $\boldsymbol{h}$ in Table E (2) and the context vector $\boldsymbol{c}$ in Table F (1) are the outputs of two different LSTMs in the language decoder, and both of them will be specified in Section A.3.

A.3. Language Decoder

As discussed in Section 3.2 of the main paper, the whole language decoder is built by stacking M single language decoders with a common structure while the parameters are different. We set the top-down LSTM [1] as our single language decoder and its architecture is shown in Table G. Specifically, for the m-th decoder, the input $\boldsymbol{i}^{m-1}$ in Table G (1) is the output of the $m-1$-th decoder. When $m=1$, this input is word embedding vector $\boldsymbol{W}_{\Sigma}\boldsymbol{s}_{t-1}$, where $\boldsymbol{W}_{\Sigma}$ is a trainable embedding matrix and $\boldsymbol{s}_{t-1}$ is the one-hot vector of the word generated at time step $t-1$. In Table G (2), the output of the second LSTM $\boldsymbol{h}_2^{t-1}$ at time step $t-1$ is used as the context vector $\boldsymbol{c}$ in Table D (1) and Table F (1), and the output of the first LSTM $\boldsymbol{h}_1^t$ in Table G (11) is used as the hidden vector $\boldsymbol{h}$ in Table E (2). After getting the output of the M-th language decoder $\boldsymbol{i}^M$, a fully connected layer and softmax activation are used for producing the word distribution $P(s)$ (cf. Section 3.3 of the main paper).

B. Human Evaluation

In the experiment (cf. Section 4.2 and Figure 5 of the main paper), we conducted human evaluation for better evaluating the qualities of the captions generated by different methods. In humane evaluation, the invited workers were required to compare the captions from two perspectives: 1) the fluency, *e.g.*, less grammar error, and descriptiveness, *e.g.*, more human-like descriptions, of the generated captions, and 2) the relevance of the generated captions to images. Figure A shows one example of the interface of our human evaluation.

Table A: The details of OBJECT module.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	RoI features	$\mathcal{R}_O$ $(N \times 2,048)$	-
(2)	(1)	FC$(\cdot)$	$\mathcal{Z}_O$ $(N \times 1,000)$	FC$(2,048 \rightarrow 1,000)$
(3)	(2)	Leaky ReLU	$\mathcal{V}_O$ $(N \times 1,000)$	-

Table B: The details of ATTRIBUTE module.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	RoI features	$\mathcal{R}_A$ $(N \times 2,048)$	-
(2)	(1)	FC$(\cdot)$	$\mathcal{Z}_A$ $(N \times 1,000)$	FC$(2,048 \rightarrow 1,000)$
(3)	(2)	Leaky ReLU	$\mathcal{V}_A$ $(N \times 1,000)$	-

Table C: The details of RELATION module.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	RoI features	$\mathcal{R}_O$ $(N \times 2,048)$	-
(2)	(1)	multi-head self-attention (Eq.(4))	$\textbf{head}_i$ $(N \times 256)$	$\boldsymbol{W}_i^1$ $(2,048 \times 256)$ $\boldsymbol{W}_i^2$ $(2,048 \times 256)$ $\boldsymbol{W}_i^3$ $(2,048 \times 256)$
(3)	(2)	multi-head vector (Eq.(5))	$\mathcal{M}$ $(N \times 2,048)$	$\boldsymbol{W}_C(2,048 \times 2,048)$
(4)	(3)	feed-forward FC$_2$(ReLU(FC$_1(\cdot)$))	$\mathcal{V}_R$ $(N \times 1,000)$	FC$_1$ $(2,048 \rightarrow 2,048)$ FC$_2$ $(2,048 \rightarrow 1,000)$

Table D: The details of FUNCTION module.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	context vector	$\boldsymbol{c}$ $(1,000)$	-
(2)	(1)	FC$(\cdot)$	$\boldsymbol{z}_F$ $(1,000)$	FC$(1,000 \rightarrow 1,000)$
(3)	(2)	Leaky ReLU	$\hat{\boldsymbol{v}}_F$ $(1,000)$	-

Table E: The details of the common structure of three attention networks.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	feature set	$\mathcal{V}$ $(N \times 1,000)$	-
(2)	-	hidden vector	$\boldsymbol{h}$ $(1,000)$	-
(3)	(2)	attention weights $\boldsymbol{w}_a \tanh(\boldsymbol{W}_v \boldsymbol{v}_n + \boldsymbol{W}_h \boldsymbol{h})$	$\boldsymbol{\alpha}$ (N)	$\boldsymbol{w}_a$ (512), $\boldsymbol{W}_v$ $(512 \times 1,000)$ $\boldsymbol{W}_h(512 \times 1,000)$
(4)	(3)	Softmax	$\boldsymbol{\alpha}$ (N)	-
(5)	(1),(4)	weighted sum $\boldsymbol{\alpha}^T \mathcal{V}$	$\hat{\boldsymbol{v}}$ $(1,000)$	-

C. More Qualitative Examples

Figure B exhibits three visualizations for explaining how RELATION module generates relation specific words. For example, in the middle figure, at the third time step, RELATION module focuses more on the "paw" part (red box) of one bird, and meantime the knowledge about "bird" (yellow box) and "tree" (blue box) is also incorporated to the "paw" part of the bird by multi-head self-attention technique (cf. Eq.(4) of the main paper). By exhaustively considering these visual clues, a more accurate action "perch" is generated.

Table F: The details of soft fusion.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	context vector	$\boldsymbol{c}$ $(1,000)$	-
(2)	-	attended object feature	$\hat{\boldsymbol{v}}_O$ $(1,000)$	-
(3)	-	attended attribute feature	$\hat{\boldsymbol{v}}_A$ $(1,000)$	-
(4)	-	attended relation feature	$\hat{\boldsymbol{v}}_R$ $(1,000)$	-
(5)	-	function feature	$\hat{\boldsymbol{v}}_F$ $(1,000)$	-
(6)	(1),(2),(3),(4)	Concatenate	$\boldsymbol{x}$ $(4,000)$	-
(7)	(6)	LSTM$_C$ $(\boldsymbol{x}; \boldsymbol{h}_C^{t-1})$	$\boldsymbol{h}_C^t$ (1,000)	LSTM$_C$ (4,000 $\rightarrow 1,000$)
(8)	(7)	Softmax	$\boldsymbol{w}$ (4)	-
(9)	(2),(3),(4),(8)	$\hat{\boldsymbol{v}} = \text{Concat}(w_O\hat{\boldsymbol{v}}_O, w_A\hat{\boldsymbol{v}}_A, w_R\hat{\boldsymbol{v}}_R, w_F\hat{\boldsymbol{v}}_F)$	$\hat{\boldsymbol{v}}$ $(4,000)$	-

Table G: The details of the single language decoder.

Index	Input	Operation	Output	Trainable Parameters
(1)	-	the output of the last decoder	$\boldsymbol{i}^{m-1}$ $(1,000)$	-
(2)	-	the output of LSTM$_2^m$ at $t-1$	$\boldsymbol{h}_2^{t-1}$ (1,000)	-
(3)	-	object feature set	$\mathcal{V}_O$ $(N \times 1,000)$	-
(4)	-	attribute feature set	$\mathcal{V}_A$ $(N \times 1,000)$	-
(5)	-	relation feature set	$\mathcal{V}_R$ $(N \times 1,000)$	-
(6)	-	function feature	$\hat{\boldsymbol{v}}_F$ $(N \times 1,000)$	-
(7)	(3)	mean pooling	$\bar{\boldsymbol{v}}_O$ $(1,000)$	-
(8)	(4)	mean pooling	$\bar{\boldsymbol{v}}_A$ $(1,000)$	-
(9)	(5)	mean pooling	$\bar{\boldsymbol{v}}_R$ $(1,000)$	-
(10)	(1),(2),(7),(8),(9)	concatenate	$\boldsymbol{u}^t$ $(5,000)$	-
(11)	(10)	LSTM$_1^m$ $(\boldsymbol{u}^t; \boldsymbol{h}_1^{t-1})$	$\boldsymbol{h}_1^t$ $(1,000)$	LSTM$_1^m$ $(5,000 \rightarrow 1,000)$
(12)	(3),(11)	attention network (Table E)	$\hat{\boldsymbol{v}}_O$ $(1,000)$	-
(13)	(4),(11)	attention network (Table E)	$\hat{\boldsymbol{v}}_A$ $(1,000)$	-
(14)	(5),(11)	attention network (Table E)	$\hat{\boldsymbol{v}}_R$ $(1,000)$	-
(15)	(2),(6),(12),(13),(14)	soft fusion (Table F)	$\hat{\boldsymbol{v}}^t$ $(4,000)$	-
(16)	(11),(15)	LSTM$_2^m$ $([\boldsymbol{h}_1^t, \hat{\boldsymbol{v}}^t]; \boldsymbol{h}_2^{t-1})$	$\boldsymbol{h}_2^t$ $(1,000)$	LSTM$_2^m$ $(5,000 \rightarrow 1,000)$
(17)	(1),(16)	add	$\boldsymbol{i}^m$ $(1,000)$	-

Figure C shows more comparisons between captions generated by CNM and Module/O. We can find that compared with Module/O, our CNM prefers to use some more accurate words to describe the appeared objects, attributes, and relations. For example, in Figure C (a), the attribute "busy" can be assigned to "street", and in Figure C (c), the action "feed" can be correctly generated.

References

[1] 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*, number 5, page 6, 2018. 1

Which caption among 3 is better, from the perspectives of language and visual

language: whether this caption contains more details, or interesting words, or summary words, or less error?

visual: is this caption related to the image?

Please pairwise compare each two captions according to descriptiveness and relevance.

Evaluation # 8 | Evaluations of Captions. Help

The captions are given as follows:

(1): a group of chefs in a kitchen preparing food

(2): a group of chefs preparing food in a kitchen

language (1) vs. (2):

A:(1) is better B:(2) is better C:two captions are similar

(1): a group of chefs in a kitchen preparing food

(3): a group of men preparing food in a kitchen

language (1) vs. (3):

A:(1) is better B:(3) is better C:two captions are similar

(2): a group of chefs preparing food in a kitchen

(3): a group of men preparing food in a kitchen

language (2) vs. (3):

A:(2) is better B:(3) is better C:two captions are similar

(1): a group of chefs in a kitchen preparing food

(2): a group of chefs preparing food in a kitchen

Visual (1) vs. (2):

A:(1) is better B:(2) is better C:two captions are similar

(1): a group of chefs in a kitchen preparing food

(3): a group of men preparing food in a kitchen

Visual (1) vs. (3):

A:(1) is better B:(3) is better C:two captions are similar

(2): a group of chefs preparing food in a kitchen

(3): a group of men preparing food in a kitchen

Visual (2) vs. (3):

A:(2) is better B:(3) is better C:two captions are similar

Next

Figure A: The evaluation interface for comparing captions generated by different models.

Figure B: Three visualizations show how RELATION module generates relation specific words like quantifiers and verbs. The red box in each image is the attended image region (with the largest soft weight) when RELATION module generates a relation specific word. The thickness of lines connecting different boxes is determined by the soft attention weights computed by self-attention technique in Eq.(4). The thicker the line connecting two boxes is, the larger the soft weight between two bounding boxes is.

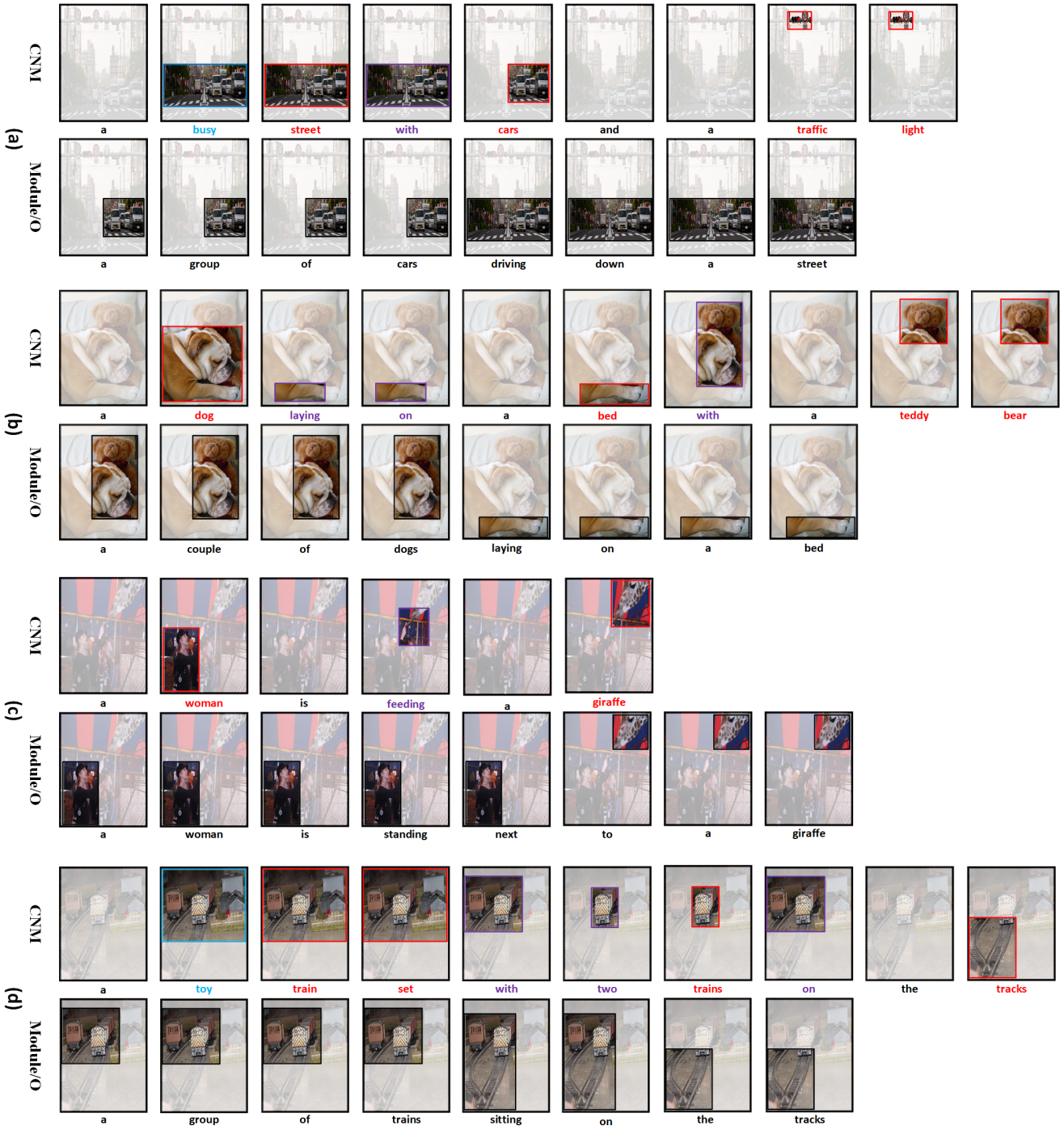

Figure C: The visualizations of the caption generation process of two methods: CNM#3 and Module/O. For CNM, different colours refer to different modules, *i.e.*, blue for ATTRIBUTE module, red for OBJECT module, purple for RELATION module, and black for FUNCTION module. For simplicity, we only visualize the module layout generated by the last module controller of the deeper decoder and only the image region with the largest soft weight is shown. For Module/O, only image region with the largest soft weight is visualized with black boundary.