Supplementary Material for
Show, Edit and Tell: A Framework for Editing Image Captions

1. DCNet

An overview of our DCNet is shown in Figure 2. We consider the existing caption as a "noisy" caption, and wish to encode it into a compressed representation and decode the compressed representation to the desired output. Notably, an LSTM-based de-noising auto-encoder is equivalent to a Sequence-to-Sequence model.

For the encoder, we encode the noisy existing caption using a bi-directional LSTM and set the dimension of each direction to 512. Specifically, an existing sequence of N words is first converted into a sequence of word vectors using an embedding layer: $\left[w_1^S, w_2^S, \ldots..w_N^S\right]$ where $w_t^S \in \mathcal{R}^{1024}$, and serve as input to a bi-directional LSTM:

$$\overrightarrow{e_t} = Bi - LSTM\left(\overrightarrow{e_{t-1}}, w_t\right) \tag{1}$$

$$\overleftarrow{e_t} = Bi - LSTM\left(\overleftarrow{e_{t+1}}, w_t\right) \tag{2}$$

Which results in a matrix $E = [(\overrightarrow{e_1}; \overleftarrow{e_1}) \ldots (\overrightarrow{e_N}; \overleftarrow{e_N})]$, where ; indicates concatenation and $E \in \mathcal{R}^{1024}$. The last hidden states of the bi-directional LSTM are concatenated and fed into a single feed-forward layer with tanh activation function:

$$E_N = \tanh\left(W_O \cdot [\overrightarrow{e_N}; \overleftarrow{e_N}]\right) \tag{3}$$

For the decoder, we use the Top-Down decoder [1] and set the dimension size to 1024. The input to the Attention-LSTM x_t^1 comprise of the current word embedding, the previous hidden state of the language LSTM and the last encoder hidden state, such that $x_t^1 = \left[w_t; E_N; h_{t-1}^2\right]$. The output of the attention LSTM is used to compute an attention vector over the textual features E:

$$c_t^e = \sum_{i=1}^{N} \alpha_{t_i} E_i \tag{4}$$

where

$$\alpha_t = \operatorname{softmax}\left(w_h^T \tanh\left(W_g E + W_k h_t^1\right)\right) \tag{5}$$

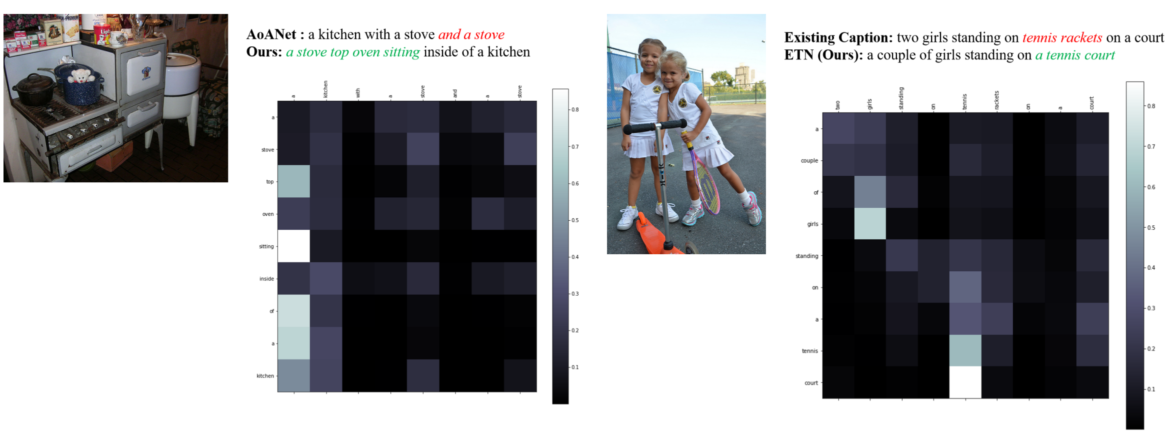

Figure 1. Visualization of the selected words from SCMA

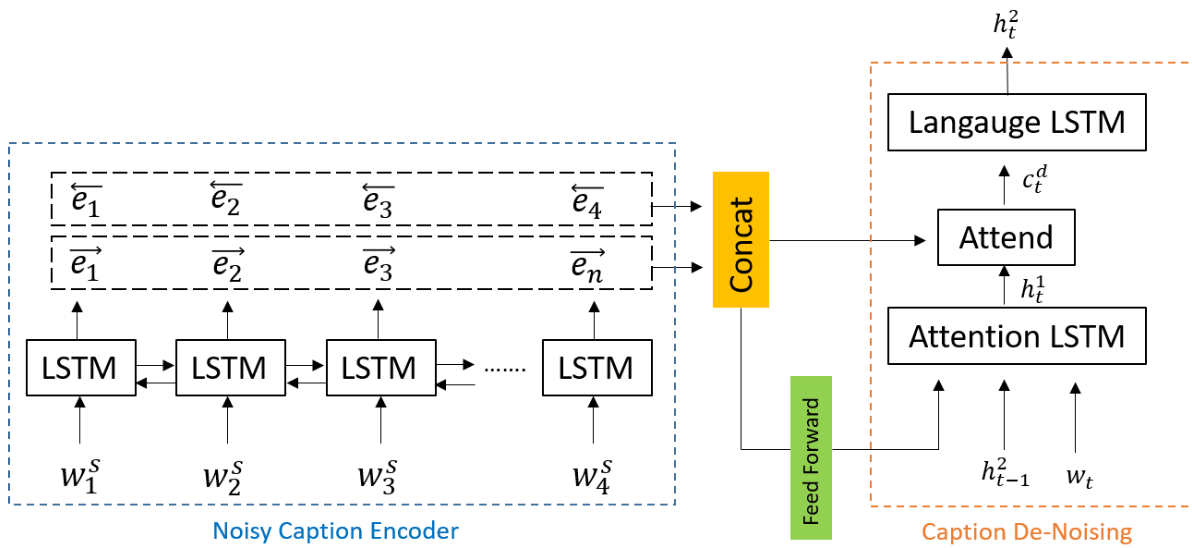

Figure 2. An overview of our DCNet sub-module, which is an LSTM-based de-noising autoencoder.

Table 1. Performace of our Single Model on the Online COCO Testing Server, where B-N, M, R, and C are short for BLEU-N, METEOR, ROUGE-L and CIDEr-D. All values are reported as percentage (%).

Model	B-1		B-2		B-3		B-4		M		R		C	
Metric	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCST [3]	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 [5]	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
StackCap [2]	77.8	93.2	61.6	86.1	46.8	76.0	34.9	64.6	27.0	35.6	56.2	70.6	114.8	118.3
Up-Down [1]	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
CAVP [6]	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 [4]	**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
ETN (Ours)	80.3	94.7	64.8	88.8	**50.2**	**79.9**	**38.3**	**69.5**	**28.6**	**37.8**	**58.4**	**73.5**	**123.6**	**125.7**

The output of the Attention-LSTM and the context vector c_t^e serve as input to the Language-LSTM, such that: $x_t^2 = [h_t^1; c_t^e]$. The output of the Language LSTM h_t^2 is then fed to the output layer which predicts a word from the vocabulary.

2. SCMA Decision Visualiztion

In this section, we include more results to visualize the decisions from the SCMA mechanism. Figure 1 shows the alignment plot between the existing caption and the generated caption.

3. Single Model Results on COCO Online Testing Server

We submitted our results evaluated on the official MSCOCO testing set to the online testing server. Table 1 shows the performance of our Single Model (which includes EditNet and DCNet) on the Online COCO Test Server. For fair comparison, we directly compare our method (ETN) with other state-of-art methods which also report the scores of their single model on the Online COCO Test Server (e.g. SGAE [4]). Note that the scores reported for SGAE are for the single model (and not ensemble).

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. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2017. 1, 2

[2] Jiuxiang Gu, Jianfei Cai, Gang Wang, and Tsuhan Chen. Stack-captioning: Coarse-to-fine learning for image captioning. In *AAAI*, 2017. 2

[3] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, 2016. 2

[4] Xu Yang, Kaihua Tang, Hanwang Zhang, and Jianfei Cai. Auto-encoding scene graphs for image captioning. In *CVPR*, 2018. 2

[5] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4904–4912, 2016. 2

[6] Zheng-Jun Zha, Daqing Liu, Hanwang Zhang, Yongdong Zhang, and Feng Wu. Context-aware visual policy network for fine-grained image captioning. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2