15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

Attention to Describe Products with Attributes

Shuohao Li
National University of Defense Technology
Changsha, China
lishuohao@vision.is.tohoku.ac.jp

Kota Yamaguchi Takayuki Okatani
Tohoku University
Sendai, Japan
{kyamagu, okatani}@vision.is.tohoku.ac.jp

Abstract

In e-commerce environment, shop owners and advertisers give descriptive details of the product to attract potential customers. Can a computer vision technique recognize and describe the details of a product in the same way? In this paper, we study how the attention mechanism benefits in product phrase generation with attributes. We present a phrase generation model consisting of convolutional neural networks, recurrent neural networks, and the attention mechanism to look into the detail of the image. We construct attribute-rich phrases from metadata in Etsy dataset that consist of an adjective, a material tag, and product category, and learn the model to describe products. Our empirical results suggest that our model improves the description quality in both machine-translation metric and human evaluation.

1 Introduction

As the image recognition performance improves year by year, researchers focus more on recognition of more diverse and structured concepts from an image, in the form of natural language [9, 5, 4, 10, 12]. Language description has the advantage over a simple label prediction in that the output naturally encodes the structure of various concepts, such as an action or a relationship between objects [6]. In this paper, we turn our attention to modifier expressions in object recognition. Modifiers, such as adjectives or prepositional phrases, have been often considered as attributes and treated in the form of multi-label classification problem [7] or adjective-noun pair prediction [1]. However, learning every possible modifier in supervised learning is unrealistic due to the explosive vocabulary size to collect data [11]. As an alternative approach, we consider attribute recognition in the context of phrase generation.

We use e-commerce data from the Etsy dataset [11] to study modifier expressions. E-commerce data have the characteristics that images often show only a single product in the center, and the description contains plenty of detailed explanation regarding the product in addition to tags and other meta-data. Unlike generating titles from noisy data [13], we attempt to learn and generate a phrase consisting of a sequence of adjective, material tag, and product category given an image. While our phrase composition could somewhat restrict the diversity of expressions, we are able to enforce the generation model to always learn and generate a modifier to the given product image without noise influence.

In this paper, we study the effect of attention mechanism in the product phrase genration. Attention mechanism has been shown to improve general image captioning [12], by assigning soft weights to the internal representation to better represent a specific object in the scene. With attention, the language generator can make more focus on which object to describe in the output word sequence. In this paper, we explore how attention works in recognizing modifiers to the product that requires attribute recognition from the details. Specifically, we study two types of attention approaches for phrase generation: one based on spatial regions, and the other based on feature channels in the deep convolutional neural network (CNN).

Our model is based on the popular combination of CNN and recurrent newtork with long short-term memory cell (LSTM) [10], with attention mechanism [12]. Attention makes a focus on specific internal representation in the deep network, and we expect the attention could better extract local information essential to recognize attributes such as material in the generated phrase. The experimental result using Etsy dataset suggests attention mechanism improves the product description in both machine translation metrics and human evaluation.

The following summarizes our contribution.

- We formulate attribute recognition as a composed phrase generation problem using the e-commerce data.
- We study the effect of attention mechanism on generating phrases with attributes.
- The empirical study using Etsy dataset shows attention makes improvement over baselines.

2 Phrase generation model

We show the overview of our model in Fig. 1. Our model consists of feature encoding by CNN, soft feature selection by attention mechanism, and feature decoding by LSTM network. We describe the overall procedure in the following. In the first feature encoding stage, we give an input image $\mathbf{I}$ to the CNN. From the CNN's internal representation, we extract two feature representations for sequence generation. One is the global feature $\mathbf{G} = \mathrm{CNN}_{\mathrm{fc}n}(\mathbf{I})$ that refers the final internal representation in the fully-connected n-th layer in the CNN. The other is the local feature $\mathbf{M} = \mathrm{CNN}_{\mathrm{conv}n}(\mathbf{I})$ that represents the feature maps of input image taken from the internal n-th convolution layer. In feature attention stage, we assign attention weights to the feature maps to extract locally-selected features for the sequence generator. Let us express the length of LSTM network by T, and the attention weights at the t-th timestep by $\mathbf{a}_t = \{a_1^t, a_2^t, \ldots a_L^t\}$, where $a_i \in \mathbf{R}, \sum_{i=1}^m a_i^t = 1$, and $t \in \{1, 2, \ldots T\}$. We obtain the local features $\mathbf{L}_t = \mathbf{a}_t \cdot \mathrm{vec}(\mathbf{M})$, and concatenate with the global feature $\mathbf{G}$ to form the feature input $\mathbf{V}_t$ to the generator at timestep t. Here, vec() refers a vectorization operator for the convolutional

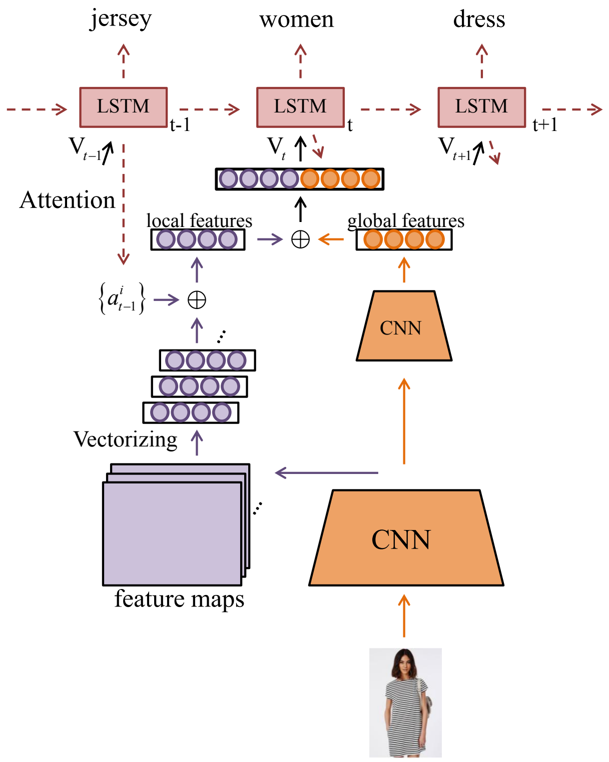
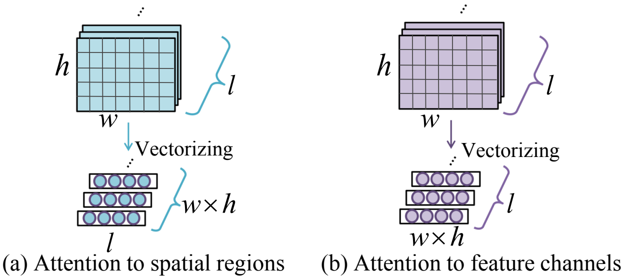


Figure 1. Our genration model consisting of CNN-LSTM network with attention mechanism.



(a) Attention to spatial regions (b) Attention to feature channels

Figure 2. Two vectorization approaches.

features $\mathbf{M}$, and we explore two different approaches of vectorization in this paper. The generator takes the attended feature input $\mathbf{V}_t$ and decodes to the target word W_t. Details of each stage are discussed below.

2.1 Image encoding by CNN

In our model, we use VGG-16 [8] for encoding global image representation and for obtaining locally attended representation. We extract two types of features from image to consider both details and the context of the image content in phrase generation. Our global feature $\mathbf{G}$ is the output of fc7 layer. In order to find out which layer contains more local features of the input image, we extract the feature maps $\mathbf{M}$ of different convolutional layers. With the attention weights described in the next section, we concatenate the global and local representation to form an input to the sequence generator.

2.2 Attention to the detail

Feature representation from the convolutional layers contain redundant or even distractor information unncessesary to recognize attributes.

The conventional attention model [12] tries to attend the spatial regions of the input image which usually contain the concerned objects. In the encoding process in this model, the values on corresponding position of all channels in feature maps are regarded as the features of the object. However, the local features for the attributes are usually extracted by a few filters in CNN. Therefore, in this paper, we also attempt to attend channels in feature maps.

As shown in Fig. 2, we evaluate two variants of the attention mechanisms based on: (a) spatial regions, and (b) feature channels in the convolutional layers, where w and h represent the width and height of $\mathbf{M}$, and l represents the number of channels. After vectorization, we can get a sequence of feature vectors $\text{vec}(\mathbf{M}) = \{S_1, S_2, ..., S_m\}$. We apply attention weights a_i^t at timestep t to obtain the local feature input $\mathbf{L}_t$ to the sequence generator:

$$\mathbf{L}_t = \sum_{i=1}^{m} a_i^t S_i \tag{1}$$

where m is the length of feature sequence $\text{vec}(\mathbf{M})$. The attention weight a_i^t is computed at each time step t inside the unit of LSTM network.

2.3 Phrase generation by LSTM

LSTM is a variant of recurrent connection that aims at addressing propagation between elements in a long sequence [3], and commonly used in various natural language tasks. We utilize LSTM to generate a product phrase. LSTM maps an input x_t, and the hidden state h_{t-1} to an updated state h_t at timestep t:

$$h_t = \text{LSTM}\left(h_{t-1}, x_t\right), \tag{2}$$

where h_0 is a constant to indicate the beginning state, $x_1 = BOS$ is a tag which represents the beginning of the sequence, and $x_N = EOS$ represents the ending of the sequence. N is the length of sequence.

The core of the LSTM unit is a memory-cell c_t and three gates i_t, o_t, f_t (see Fig.3). We describe the LSTM behavior in the Eq. 3:

$$\begin{aligned}
x_t &= \mathbf{W}_e \mathbf{V}_t, \\
i_t &= \sigma\left(\mathbf{W}_{ix} x_t + \mathbf{W}_{im} h_{t-1} + b_i\right), \\
f_t &= \sigma\left(\mathbf{W}_{fx} x_t + \mathbf{W}_{fm} h_{t-1} + b_f\right), \\
o_t &= \sigma\left(\mathbf{W}_{ox} x_t + \mathbf{W}_{om} h_{t-1} + b_o\right), \\
g_t &= \varphi\left(\mathbf{W}_{gx} x_t + \mathbf{W}_{gm} h_{t-1} + b_g\right), \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t, \\
h_t &= o_t \odot \varphi\left(c_t\right),
\end{aligned} \tag{3}$$

where the weight matrices is denoted by $\mathbf{W}_{ij}$ and biases b_j, which are the trainable parameters. $\odot$ represents the element-wise product. Memory-cell c_t encodes the information of previous memory-cell c_{t-1} and current input. The gates control the flow of information in LSTM units. $\mathbf{V}_t$ is the feature vector concatenating global features and local features, and $\mathbf{W}_e$ is

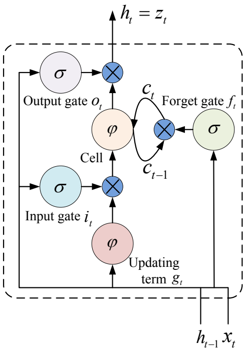


Figure 3. LSTM cell [3]. The memory cell is the core of the LSTM unit and modulated by the input, output and forget gates controlling how much information is transferred at each step.

the embedding parameter which maps the feature vector to the same space as LSTM unit. σ is the sigmoid, and φ is the hyperbolic tangent non-linearity.

The final step in predicting a distribution $p\left(W_t \left|\mathbf{V}_t\right.\right)$ at step t is to take a softmax over the output h_t of the unit, and produce a distribution over the character space $\mathbf{S}$ of possible per-timestep outputs:

$$p\left(W_t = s\left|\mathbf{V}_t\right.\right) = \frac{\exp\left(\mathbf{W}_{zs}h_{t,s} + b_s\right)}{\sum\limits_{s^{,}\in\mathbf{S}}\exp\left(\mathbf{W}_{zs}h_{t,s^{,}} + b_s\right)}. \quad (4)$$

In LSTM network, the input of the t-th unit is the feature vector $\mathbf{V}_t$. Eq.1 shows how this vector is generated from feature sequence.

In Eq.1, the attention weights a_i^t reflects the relevance of S_i in the feature sequence, and a_i^t is dependent on all previous words $\{W_1, W_2, \dots W_t\}$ generated by LSTM network. In LSTM network, the previous hidden state h_{t-1} can summarizes the information of all previous generated words. Hence, we take h_{t-1} as the input to define the attention weights:

$$a_i^t = \frac{\exp\left\{e_i^t\right\}}{\sum\limits_{j=1}^{m}\exp\left\{e_j^t\right\}}, \quad (5)$$

$$e_i^t = \mathbf{w}^T\tanh\left(\mathbf{W}_a h_{t-1} + \mathbf{U}_a S_i + \mathbf{b}_a\right). \quad (6)$$

At each time step, we update the weights a_i^t of the attention model, and get the attended feature vector $\mathbf{L}_t$.

Our attention model has parameters $\mathbf{W}_{ij}$, b_j, $\mathbf{W}_e$, $\mathbf{W}_{zs}$, b_s, $\mathbf{w}^T$, $\mathbf{W}_a$, $\mathbf{U}_a$ and $\mathbf{b}_a$ from Eq.3, 4 and 6. These parameters are learned all together from training data. Our model is trained in an end-to-end manner by minimizing the following penalized negative log-likelihood, which directly indicates the loss value between labels and generated words. The loss function is defined as follows:

$$l\left(\mathbf{X}\right) = \sum_{\{\mathbf{I}_i,\mathbf{C}_i\}\in\mathbf{X}}\sum_{t=1}^{N} -\log p\left(W_t = \mathbf{C}_{i,t}\left|\mathbf{V}_t\right.\right), \quad (7)$$

where $\mathbf{X}$ represents the dataset, $\mathbf{I}_i, \mathbf{C}_i$ represents the input image and sequence pair, and $\mathbf{C}_{i,t}$ represents the t-th word of the i-th sentence. We train our model using stochastic gradient descent (SGD) with momentum, and backpropagation is used to compute the gradient of parameters. We also use the ADADELTA to calculate per-dimension learning rates, instead of setting learning rate manually.

3 Experiments

3.1 Dataset and pre-processing

We build the dataset of phrases for e-commerce products from the Etsy dataset [11]. Similarly to [11], we select products under the clothing category for our evaluation and applied near-duplicate removal based on meta-data, which resulted in 104,979 images. To further remove duplicate images in the dataset, we apply near-duplicate removal based on image features. We used the pre-trained VGG-16 model to extract the fc7 feature of the image and removed the images having close features. After cleansing, our evaluation data resulted in 88,661 images in total, and we split them into 59,104 images for training, 14,781 images for validation, and 14,776 images for testing.

Every image in the dataset has a description, a category, a price, and material tags. We apply syntactic analysis [2] to extract part-of-speech (POS) for each word in the description. We select 250 most frequent adjectives from all parsed words. Then we build a sequence of random combination of adjective, material tag, and product category as a phrasal description of the image, such as “black cotton women dress”, or “yellow leather women shoes”. We discard phrases whose length is greater than 10 and prepare 5 phrases for every image in the dataset.

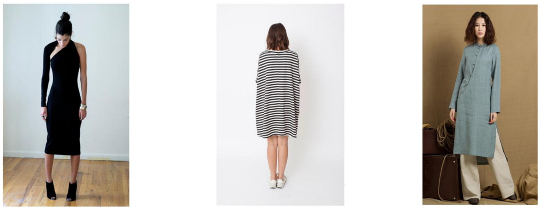
3.2 Evaluation

We compare the following models.
CNN-LSTM: Combination of CNN + LSTM [10].
Att: CNN-LSTM with attention to the global feature [12].
S-attN: Our spatial attention at Nth layer.
C-attN: Our channel attention at Nth layer.
We evaluate the above models using machine translation metrics and also by human evaluation.

For machine translation metrics, we use BLEU-{1,2,3,4}, Meteor and CIDEr metrics to evaluate the quality of the generated descriptions. Table 1 summarizes the results on 5,000 test images. We find that **C-att5** shows the best performance, and **S-att4** and **C-att4** follow. Interestingly, **Att** shows the worst performance. One reason could be that the original attention model was designed to describe the relationships between different objects, whereas our product images contain only one object and the global attention does not work.

Fig. 4 shows a few examples for our **C-att5** and baselines. Red phrases represent somewhat inaccurate descriptions. The generated phrase looks similar in general (left image), but our model tends to make less mistakes (center and right).

We visualize the attended regions in input images in Fig. 5 to analyze how different attention works in our model. Fig. 5a is the visualization of **C-att5** and

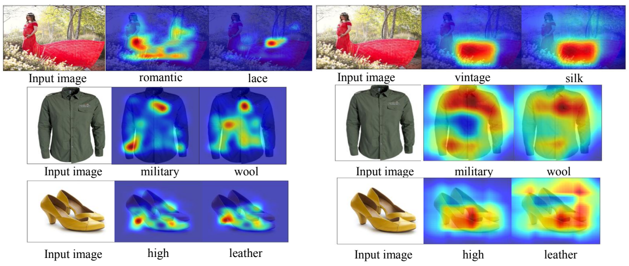


Our: black jersey women dress
CNN-LSTM: black jersey women dress
Att: black jersey women dress

Our: long viscose women dress
CNN-LSTM: black knit women dress
Att: short line women dress

Our: long linen women dress
CNN-LSTM: long linen women jacket
Att: black jersey women dress

Figure 4. Selected generation examples.



(a) Visualization on channel attention model

(b) Visualization on spatial attention model

Figure 5. Visualization of attention weights.

Table 1. Machine translation metrics.

Model	CNN-LSTM	Att	S-att4	S-att5	C-att4	C-att5
BLEU-1	0.406	0.381	**0.411**	0.404	0.405	**0.411**
BLEU-2	0.286	0.260	*0.286*	0.282	0.285	**0.291**
BLEU-3	0.213	0.191	0.212	0.209	*0.213*	**0.218**
BLEU-4	0.161	0.142	0.16	0.156	*0.162*	**0.167**
Meteor	0.189	0.176	0.19	0.189	*0.191*	**0.192**
CIDEr	0.944	0.848	0.94	0.935	**0.963**	*0.957*

Table 2. Crowd voting counts

C-att5	CNN-LSTM	None	Invalid
2055	1832	1094	9

5b is **S-att4**. We find that attended regions in **C-att5** are narrower than **S-att4**. The attention from **C-att4** shows focus on a specific part, such as "high" for heels. For spatial attetion, we directly show the weights of the regions in feature maps at conv5. For channel attention, we show the channel-weighted sum of feature maps at conv5. Channel attention seems harder to interpret, and it is our future work to develop a better visualization technique.

3.3 Human evaluation

We also had subjective evaluation using crowdsourcing. We randomly select 1,000 images from the test set and generate a phrase from our **C-att5** and the baseline **CNN-LSTM** model. We asked 5 workers in Amazon MTurk to select a better description of the two for every image. If all descriptions are not suitable for the given image, we allowed the worker to choose None. Table 2 shows the voting statistics. Our model obtained more votes than the baseline model.

4 Conclusion

In this paper, we present a phrase generation model with attention mechanism for describing a product image. Our model utilizes the combination of CNN, LSTM, and attention mechanism to make local feature selection to recognize details. We examined two variants of atttention mechanisms and showed that both performed better than the baselines. The channel-attention variant seems slightly outperforming the spatial attention.

5 Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 15H05919 and 16H05863

References

[1] Damian Borth, Rongrong Ji, Tao Chen, Thomas Breuel, and Shih-Fu Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *Multimedia*, pages 223–232. ACM, 2013.

[2] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

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

[4] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137, 2015.

[5] 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. *TPAMI*, 35(12): 2891–2903, 2013.

[6] 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*, pages 740–755, 2014.

[7] Wanli Ouyang, Hongyang Li, Xingyu Zeng, and Xiaogang Wang. Learning deep representation with large-scale attributes. In *ICCV*, pages 1895–1903, 2015.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[9] Yoshitaka Ushiku, Tatsuya Harada, and Yasuo Kuniyoshi. Understanding images with natural sentences. In *Multimedia*, pages 679–682. ACM, 2011.

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

[11] Sirion Vittayakorn, Takayuki Umeda, Kazuhiko Murasaki, Kyoko Sudo, Takayuki Okatani, and Kota Yamaguchi. Automatic attribute discovery with neural activations. In *ECCV*, pages 252–268. Springer, 2016.

[12] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2(3):5, 2015.

[13] Takuya Yashima, Naoaki Okazaki, Kentaro Inui, Kota Yamaguchi, and Takayuki Okatani. Learning to describe e-commerce images from noisy online data. *ACCV*, 2016.