Attention Analysis in Caption Generation

Marki Shozu
Graduate School of Humanities and Sustainable Science
Osaka Prefecture University
Osaka, Japan
sya01133@edu.osakafu-u.ac.jp

Hidekazu Yanagimoto
Graduate School of Humanities and Sustainable Science
Osaka Prefecture University
Osaka, Japan
hidekazu@kis.osakafu-u.ac.jp

***Abstract*—Caption Generation is one of the fundamental tasks combining computer vision and natural language processing. To achieve this goal, neural networks are employed to implement a caption generation system. In this paper, we proposed a caption generation system combining a CNN-based object detection system and a language model with a recurrent neural network. Especially, a vector which is sent from the object detection system to the language model is generated using an attention mechanism. Attention visualization can help us to understand the system focuses on a part of the input image in generating a caption. In the experiments, we evaluate the performance of the proposed system and discuss the effects of the attention mechanism in the image caption. Especially, the attention contributes to the improvement of caption generation but the attention is uncorrelated to system interpretation.**

Keywords—Deep Learning, Multimodal Learning, Caption Generation

I. Introduction

Caption generation is one of cross-media understanding tasks to need image understanding and text understanding. So we need image understanding, which extracts information on objects from a raw image, and text understanding, which generates a caption describing the content of the image appropriately. Many researchers develop a caption generation system combining a convolutional neural network (CNN), a recurrent neural network (RNN), and attention mechanism. CNN generates intermediate representation, which has information on input data and RNN generates a sequence of words from the intermediate representation. Usually, the intermediate representation is constructed with two methods. One approache makes the representation from the final features in CNN[1]. Another approach makes the reresentaion from intermediate featurs in CNN[2]. Especially, the later approach is realized with an attention mechanism, which generates an intemediate representation with a weighted sum of all image features. The attention mechanism is regarded as word alighnment in machine translation and focusing point in image processing. Xu et al.[2] insisted that the attention mechanism can improve caption generation performance and illustrated attention visualization. On the other hand, it was reported that the alignment did not improve the final translation accuracy[5][6] in machine translation. In this paper, we check whether the attention works well in generating a caption or not.

We found that the atttetion improved caption generation accuracy. but it did not always focus on appropriate image parts. The result means that the attention cannnot help us to understand how the system works. So in multimodal learning, attention has the same characteristics as in machine translation.

II. Related Works

Neural network based caption generation is developed by Vinyals et al.[1]. They combined a convolutional neural network and a recurrent neural network and realized the first caption generation system. The system based on a sequence-to-sequence model[3]. The convolutional neural generates a feature vector, which includes all information on an input image. Especially, a pre-trained convolutional neural network was used in the system and a language model with LSTM was used. An attention mechanism was proposed by Bahdanau et al.[4] in machine translation. The attention mechanism improves machine translation accuracy. The attention mechanism was applied to the caption generation by Xu et al.[2]. They illustrated how it generates nouns in a caption from an image. Some researchers reported that word alignment did not contribute to translation improvement directly[5][6].

III. Caption Generation with Attention Mechanism

We proposed an image caption generation system using an attention mechanism. In Fig. 1, we show a workflow of the proposed sytem. The proposed system consistes of a CNN-based feature extraction module, an RNN-based text generation module, and an attention mechanism module.

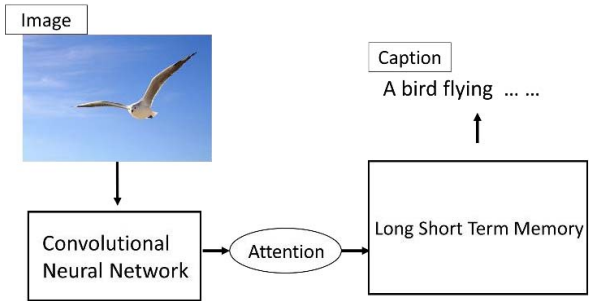

Fig. 1. The architecture of the proposed method

A. *CNN-based image feature extraction module*

In caption generation, we have to extract features from raw images. In this study, we employ a pretrained convolutional

978-1-7281-2627-2/19/$31.00 ©2019 IEEE
DOI 10.1109/IIAI-AAI.2019.00029

neural network because we need much training time and numerous amounts of training data. We employ VGG16[7] as the pretrained convolutional neural network. VGG16 has been trained with ImageNet dataset already and can classify images into 1000 object classes. We think VGG16 can extract enough features from an input image and the features are useful to generate a caption of an image. The architecture of VGG16 includes 13 convolutional layers and 3 fully-connected layers. We can divide VGG16 into two modules: a feature extraction module which consists of 13 convolutional layers and a classification module which consists of 3 fully-connected layers. In Fig. 2, we show the architecture of VGG16.

We use the output of the 13th layers in VGG16 as a feature vector of an input image because the output is the final output from a feature extraction module. Moreover, the output of a convolutional layer is assigned to a part of an image easily. So, we can discuss how a part of an image affects the final output easily. The size of an image feature vector $\boldsymbol{I_i}$ is 512 x 14 x 14.

$$\boldsymbol{I_i} = \mathbf{CNN}(\mathbf{image}) \qquad (1)$$

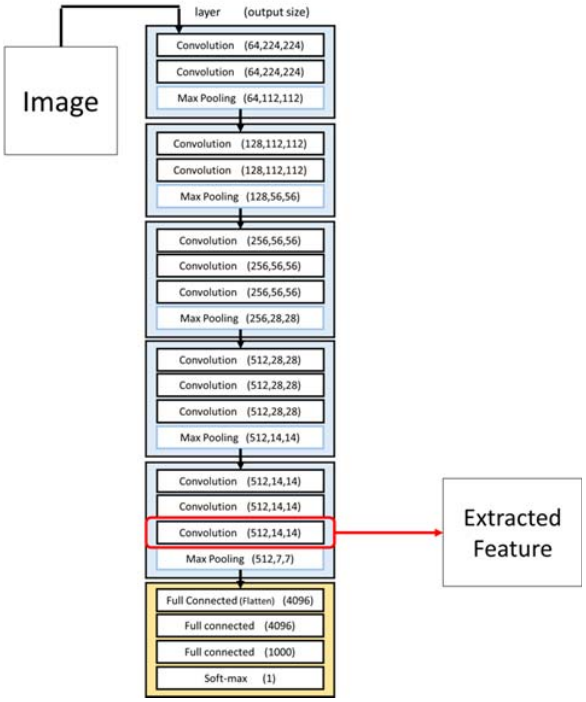

Fig. 2. The architecture of VGG16

B. Text generation module with LSTM

A text generation module is constructed with Long Short-term Memory (LSTM), which can capture long-term dependency in a sentence. In Fig. 3, we show the text generation module, which includes a word embedding layer, a recurrent layer, and an output layer. The word embedding layer transforms a 1-of-N vector into a continuous vector. Speaking concretely, the embedding layer is implemented as a matrix and the matrix is trained with training data.

$$\boldsymbol{x_t} = \mathbf{W}_e \boldsymbol{s_t} \qquad (2)$$

, where $\boldsymbol{s_t}$ is a 1-on-N vector for an input word, $\boldsymbol{x_t}$ is a embedded word vector, $\mathbf{W}_e$ is an embedding matrix.

LSTM has some parameters, a hidden stat, and a cell state.

$$\boldsymbol{m_t} = \mathbf{W}_x \boldsymbol{x_t} + \mathbf{W}_h \boldsymbol{h_{t-1}} + \mathbf{W}_z \boldsymbol{z_t} \qquad (3)$$

$$\boldsymbol{h_t}, \boldsymbol{c_t} = \mathbf{LSTM}(\boldsymbol{m_t}, \boldsymbol{c_{t-1}}) \qquad (4)$$

, where $\boldsymbol{h_t}$, $\boldsymbol{c_t}$ is a hidden state and a cell state in LSTM. The hidden state and the cell state is updated with $\boldsymbol{m_t}$ and the cell state $\boldsymbol{c_{t-1}}$. $\boldsymbol{m_t}$ is produced by the embedded word vector $\boldsymbol{x_t}$, the previous hidden state $\boldsymbol{h_{t-1}}$, and the context vector $\boldsymbol{z_t}$. The context vector $\boldsymbol{z_t}$ is made by an attention mechanism. $\mathbf{W}_x$, $\mathbf{W}_h$, $\mathbf{W}_z$ are trained parameters.

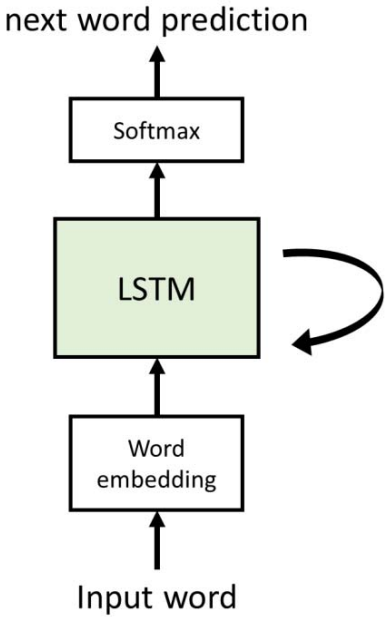

Fig. 3. The architecture of LSTM

The output, $\boldsymbol{p_t}$ is generated with the hidden state of LSTM. The $\boldsymbol{p_t}$ denotes the next word probability and the next word is determined depending on the probability.

$$\boldsymbol{p_t} = \mathbf{softmax}(\mathbf{W}_o \boldsymbol{h_t}) \qquad (5)$$

The LSTM is initialized with image feature vectors.

C. *Attention Mechanism*

We construct information which is sent from the feature extraction module to the text generation module with an attention mechanism. The information is represented as a continuous vector which is constructed as a weighted sum of feature vectors from an image. The weight is determined with an attention mechanism. In the attention mechanism, attention weights are calculated according to image feature vectors and a state in the text generation module. The following formulation denotes how to calculate the attention weights.

$$\boldsymbol{e_{ti}} = \mathrm{f_{att}}(\boldsymbol{I_i}, \boldsymbol{h_{t-1}}) \qquad (6)$$

$$\boldsymbol{\alpha_{ti}} = \frac{\exp(\boldsymbol{e_{ti}})}{\sum_{k=1}^{L} \exp(\boldsymbol{e_{ti}})} \qquad (7)$$

,where the attention weight α_{ti} is computed by an attention model $\mathrm{f_{att}}$ using multi-perceptron conditioned on the previous LSTM hidden state. Applied with Equation (2), α_{ti} is interpreted as the probability. $\boldsymbol{I_i}$ is an image feature to produce a word. We calculate a context vector z_t sent from the CNN-based module to the recurrent-based module below.

$$\boldsymbol{z_t} = \sum_{i=1}^{L} \boldsymbol{\alpha_{ti}} \boldsymbol{I_i} \qquad (8)$$

The attention mechanism is constructed with a neural network and trained with the training dataset. In this study, we do not need any other information except images and their captions.

D. *Attention Visualization*

We visualize attention weights because we check how the attention mechanism woks. Generally, attention is a continuous vector and it is difficult to combine a score in the vector with a part of an input image. Using the visualization mechanism, we can understand the relation between image features and words.

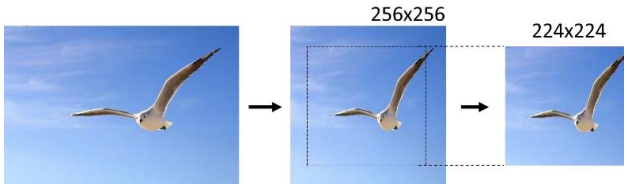

Fig. 4. Process of resizing an original image

VGG16 consists of some convolution layers and pooling layers and accept only 224x224 images. Because images for caption generation has different image size, we have to resize the image to 224x224. At first, we resize an original image to a 256x256 image and crop a 224x224 center image from the resized image. In all convolution layers, an output image has the same size as an input image and the image size is not changeable. On the other hand, 4 pooling layers in VGG16 have 4x4 window size and shrink image size. After applying VGG16 to an input image, we get a 14x14 image feature. Attention is calculated to the 14x14 image feature and we get 14x14 attention weights. In order to visualize the attention weights, we simply upsample the attention weights to a 224x224 matrix. Then we apply a Gaussian filter to the matrix and visualize the attention weights.

Only attention visualization is not understandable because it is not clear where the attention emphasizes the input image. In visualization, we superimpose the attention visualization on the input image. So we can understand some image features contribute to word generation in a caption. Fig. 6 are examples of the final visualization.

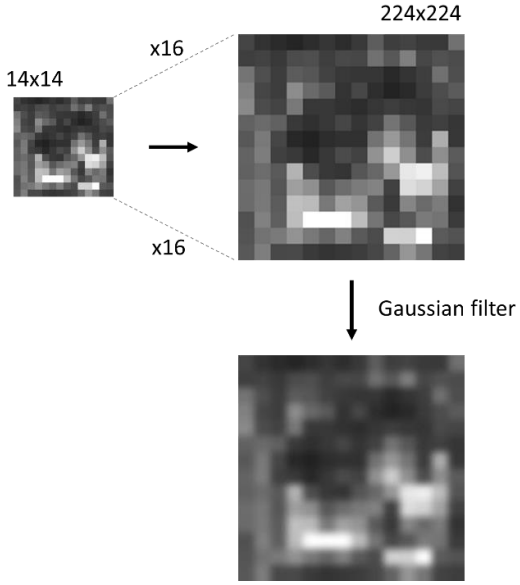

Fig. 5. Process of visualizing attention

IV. EXPERIMENTS

A. *Dataset*

In the experiments, we use Microsoft COCO dataset which has 123,287 images and 4 or 5 sentences per an image. This dataset is divided into a training set (82,783 images) and a validation set (40,504 images). There are 8,793 words in all captions in the dataset, and we neglect the words which appear less than 5 times in the training set.

B. *Settings*

The parameters of the attention model and LSTM decoder are trained by a backpropagation algorithm. Especially, we use Adam algorithm. We use the sum of the negative log likelihood of the correct word at each step as a loss function.

In Table 1, we show the hyper-parameters of this experiment .

C. *Results and Discussions*

In table 2, we show the results of evaluation by Accuracy / BLEU metrics. There was no much difference between the

results of the system without the attention mechanism and the system with the attention mechanism. However, we found that the system with the attention mechanism reached the highest accuracy (5 epoch) faster than the system without attention mechanism (12 epoch).

Table 1. Hyper parameter settings

Hyper parameter	Size
Image feature vector size	512
Embedded word vector size	512
LSTM hidden state size	512
Minibatch size	64

Table 2. Quantitative Evaluation

Model	Accuracy	BLEU
NICG (Neural Image Caption Generator)	47.3	44.3
NICG with attention mechanism	47.5	45.1

In Fig. 6, we show some examples of attention visualization. Each image in the examples denotes an attention distribution over the input image to generate a word. At first, our system paid attention to the traffic light in the input image. However, the other attentions did not work well because the attention was not able to capture "signpost" in the input image. On the other hand, the caption is generated appropriately. This result means that the attention accuracy and the caption accuracy are not correlated. We didn't use a quantitative method to estimate the attention accuracy.

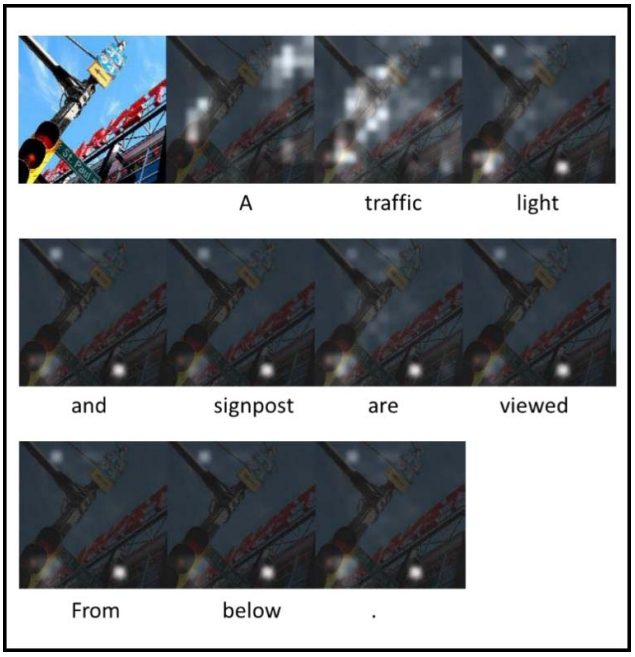

Fig. 6

In caption generation, which is one of multimodal learning approaches, the attention accuracy is uncorrelated to the caption accuracy directly. This characteristic is similar to a result in machine translation. The ordinary attention mechanism does not help us to understand how a caption generation system generates a word.

V. Conclusions

In this paper, we propose an attention based caption generation system using neural networks and we visualize the attention weights of this system. Moreover, we analyze attention using the attention weight visualization. We found that the attention-based caption generation system improved a caption of an input image but attention accuracy was uncorrelated to caption accuracy. It means that attention is not a simple alignment between a part of the image and a word in the caption.

In future works, we will discuss attention in detail analyzing more caption generation results. For example, we increase learning epochs and discuss how the attention changes according to the learning epoch. Moreover, we will discuss attention according to types of input images. For example, natural images or artificial images.

References

[1] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, "Show and Tell: A Neural Image Caption Generator," Proc. of 2015 IEEE Conference on Computer Vision and Pattern Recognition, pp.3156-3164, 2015

[2] K. Xu, J. Ba, R. Riros, K. Cho, A. Couville, R. Zemel, and Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," PMLR2015, pp.2058-2057, 2015.

[3] I. Sutskever, O. Vinyals, Le QV (2014), "Sequence to Sequence Learning with Neural Networks," NIPS, 2014, pp.3104-3112

[4] Bahdanau D, Cho K (2016), "Neural Machine Translation by Jointly Learning to Align and Translate," ICLR, 2015

[5] A. Fraser and D. Marcu, "Measuring Word Alignment Quality of Statistical Machine Translation, Computational Linguistics," Vol. 33, No. 3, pp.293-303, 2007

[6] M. Loung, H. Pham, C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," Proc. of EMNL2015, pp.1412-1421, 2015.

[7] Simonyan K, Zisserman A (2014), "Very Deep Convolutional Networks for Large-Scale Image Recognition," ICLR, 2015, pp.1-14