Image Caption Generation With Adaptive Transformer

Wei Zhang, Wenbo Nie, Xinle Li and Yao Yu

May 28, 2019

Image Caption Generation With Adaptive Transformer

1st Wei Zhang
School of Automation and Electrical
University of Science and Technology Beijing
Beijing, China
2033329616@qq.com

2nd Wenbo Nie
School of Automation and Electrical
University of Science and Technology Beijing
Beijing, China
969185900@qq.com

3rd Xinle Li
School of Automation and Electrical
University of Science and Technology Beijing
Beijing, China
xinle_li@yeah.net

4th Yao Yu*
School of Automation and Electrical
University of Science and Technology Beijing
Beijing, China
yuyao@ustb.edu.cn

***Abstract*—Encoder-decoder framework based image caption has made promising progress. The application of various attention mechanisms has also greatly improved the performance of the caption model. Improving the performance of every part of the framework or employ more effective attention mechanism will benefit the eventual performance. Based on this idea we make improvements in two aspects. Firstly we use more powerful decoder. Recent work shows that Transformer is superior in efficiency and performance to LSTM in some NLP tasks, so we use Transformer to substitute the traditional decoder LSTM to accelerate the training process. Secondly we combine the spatial attention and adaptive attention into Transformer, which makes decoder to determine where and when to use image region information. We use this method to experiment on the Flickr30k dataset and achieve better results.**

***Index Terms*—image caption, adaptive attention, transformer**

I. Introduction

Deep learning has made promising progress in the field of computer vision and natural language processing. Image caption is the cross-direction of these two fields, which has been widely investigated in the past few years [1], [2], [3], Nowadays automatically generating image description is still a challenging task. This not only needs to recognize the objects, attributes and their relationships of the image correctly, but also to generate fluent sentences. This technology can be used in many ways, such as increasing children's interest in early education, or in the social and security fields.

Image caption is a sequence modeling problem essentially, which is compatible with machine translation [4]. In translation task, the RNN-based encoder-decoder framework is usually used, which maps the source language to the target language. In image caption task, the encoder should extract the image feature to obtain a context vector, then put it into decoder to generate the language description. Convolutional neural networks are widely used in computer vision due to the excellent feature extraction capabilities. Therefore, in this task we employ convolutional neural network to extract image feature, here we use ResNet [5]. In the decoder part, Long Short Term Memory, LSTM is very common in use, this network can solve the problem of long-term dependency of sequence effectively, but due to the structure of the LSTM or other RNNs, the current output depends on the hidden state at the previous moment, so they can only operate in time steps, this makes it impossible to effectively parallelize. Ashish et al. propose Transformer [6] model to solve the parallelism problem. There is no sequence dependence on this model, it is entirely based on attention mechanism, so Transformer can run in parallel during the training phase, moreover it outperforms LSTM slightly in machine translation, therefore we employ Transformer as decoder to generate captions.

The introduction of attention mechanism [7] has greatly improved the performance of sequence modeling in natural language processing. The attention mechanism is able to align the words between encoder's inputs and decoder's outputs, and it also can memorize their correlation. Attention is a weighted average process, so it is compatible for image caption task. Xu et al. in [8] employ the soft-attention to align the regions of input image to the caption words, which improves the performance greatly. However, the feature of the image is used in the same way at every timestep when the words sequence is generated. In fact when it comes to some words which have little relevance to image, such as prepositions or conjunctions, the model should depend on the generated words to predict the next word, rather than use the image information. Lu et al. propose the adaptive attention mechanism [9] to solve this problem, in this model the decoder uses a visual sentinel to determine when to use the image feature. In order to make the advantage of attention mechanism, we combine the spatial attention and the adaptive attention into the transformer.

The contributions of this paper are as follows:

- We use Transformer as decoder to train the caption model in a parallel way, therefore improving the training efficiency of the model.

- We employ both spatial and adaptive attention into Transformer, and introduce a Scaled Dot-Product Spatial Attention and Adaptive Multi-Head Attention.
- Conduct experiments to compare the performance between LSTM and Transformer based caption models.

II. Related Work

The problem of image caption has been studied for a long time, it was originally used in traffic scene to describe the traffic conditions of vehicles. Recently, with the development of machine learning and deep learning, researchers have proposed many methods to fuse the multi modal feature between image and text effectively. The general methods can be divided into three categories: (1) Retrieval-based caption, (2) Template-based caption and (3) Novel Image Caption generation.

(1) Retrieval-based caption

Many work view the image caption as a retrieval problem, [10] propose a retrieval model, this model firstly extracts the feature of the input image, then retrieves similar images in the database, finally uses the most matching image's caption, or combine multiple similar images' caption to obtain the output. However, there are certain problems on this method, the description of the input image in this method is to reuse the data in the database, thus the scale of database determines the captions' quality. If the database is large enough to contain all kinds of images, the method can achieve good results, when it comes to rare images which are not in database will get worse results. Since the generated process is combination and reuse of similar images' captions in the database, so this method can't generate new descriptions.

(2) Template-based caption

When describing an image, people tend to analyze the objects, attributes and relationships in the image, then express these information in fluent language. Inspired by this idea, Farhadi et al. in [11] use the <object, action, scene> tree tuple as the semantic representation of the image, then fill these words into the language templates to generate the complete sentence. These templates are hand-designed, which leave the positions of the subject and the object empty, the empty positions will be filled by the key information from image. This method is able to guarantee the generated descriptions are grammatical, but the templates need to design manually, therefore the maintenance cost is relatively high, also the generated sentences pattern is relatively simple.

(3) Novel Image Caption generation

Currently, neural network based encoder-decoder framework is the most popular method, this method views image caption as image translation task. Firstly, the encoder is used to detect the objects, attributes, scenes and activities in the image, and then use a feature vector to represent these information, finally put the feature vector into the decoder to generate image descriptions. The m-RNN and NIC models are earlier work which use CNN-RNN encoder-decoder framework. Xu et al. introduce the spatial attention [8] into this framework, this attention mechanism makes the model know which image region is more important in the process of generating sentences, which boosts the performance in this task. Later, in [9] Lu et al. propose an adaptive attention, this attention mechanism makes the model know when to use the image feature in the process of generating sentences, the model can decide whether to use the image feature entirely. When generating a word that has low correlation to image, the model is able to use the language model's ability, which predicts the next word is based on the generated sentences, this method can improve the accuracy of the generated sentences. In addition to spatial attention, [12] uses semantic attention to attend the semantic information in the image to enhance the encoder's ability of extracting semantic information.

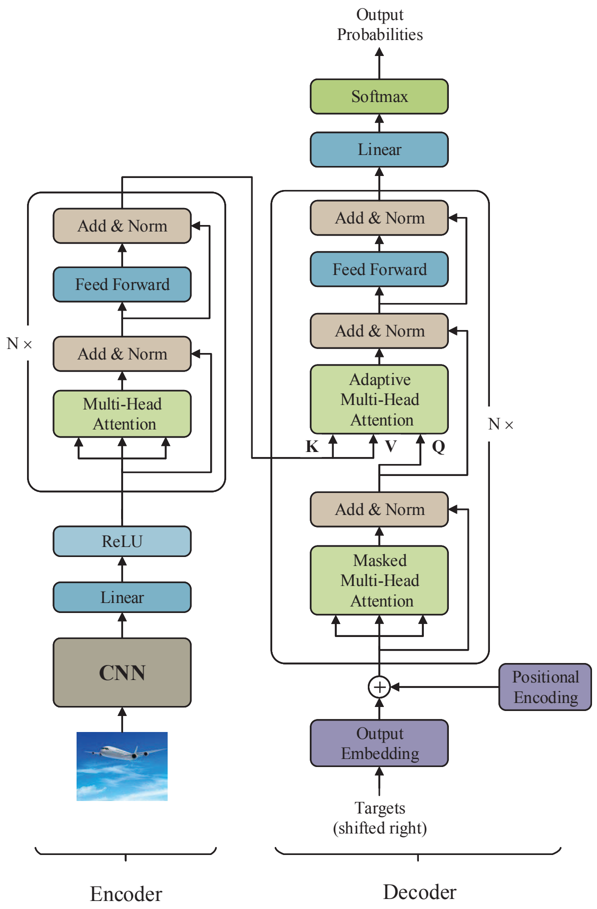


Fig. 1: Our Adaptive-Trans model, the left is CNN based encoder, the right is decoder which is a Transformer with adaptive attention.

In the encoder-decoder framework, most model is end-to-end, the output of the encoder can't be explained in specific meaning, Qi Wu et al. [13] use the high-level semantic concept as the output of the encoder, thus making the intermediate information of the model be interpretable.

In addition to the introduction of spatial attention and semantic information, Dense Caption [14] describes the multiple regions of the image, and Reinforcement Learning [15] optimizes the evaluation metric directly rather than minimizes the CrossEntropy. These methods all achieve good performance.

III. Method

Image caption is the translation from image to text, so it can be viewed as a sequence modeling problem to solve. RNN-RNN based encoder-decoder framework is used to deal with the sequence modeling problems, but image itself has no sequence information, furthermore CNN performs better than RNN in vision task. Here we use the CNN as encoder to extract image feature, the output of the encoder is a context vector which contains the necessary information from image, then put it into Transformer to generate the captions.

In the process of image caption generation, the feature of the image is input to decoder at every timestep, but not every word is closely related to the image, these words can be inferred from the previous generated sentence. In order to balance the influence between image information and generated sentences, we combine the spatial and adaptive attention into Transformer, which is dubbed as Adaptive-Trans, this model can learn to determine where and when to use image feature. The overall framework of Adaptive-Trans is shown in Fig. 1.

A. Encoder-Decoder framework

There are two parts in image caption, encoder and decoder. We employ convolutional neural network ResNet as the encoder to obtain context vector of the image, then put this vector into decoder Transformer to generate captions.

It is generally considered that the higher layer of CNN network tend to extract high level semantic, thus the output of the fully connected layer is able to represent the global information of the image. But this output lacks some spatial information, in order to utilize the spatial information, we use the output of the last convolutional layer same as Soft-Attention [8], set this vector as $V = \{V_1, \ldots, V_{H\times W}\}$, $V_i \in R^c$, where H, W and C correspond to height, width and depth of this spatial feature map respectively. The higer layer in CNN has larger receptive field, so every point in spatial feature map corresponds to a region of original image, thus the spatial vector V corresponds to $H \times W$ regions in original image.

In order to match the dimensions between CNN and Transformer, we add a linear layer and an activation function ReLU to the CNN, it can map V_i to V' from C dimension to d_{model} dimension, where d_{model} is the dimension of word embedding and intermediate state of Transformer. This vector is the spatial representation of the image, thus decoder can use it to attend to salient objects in image.

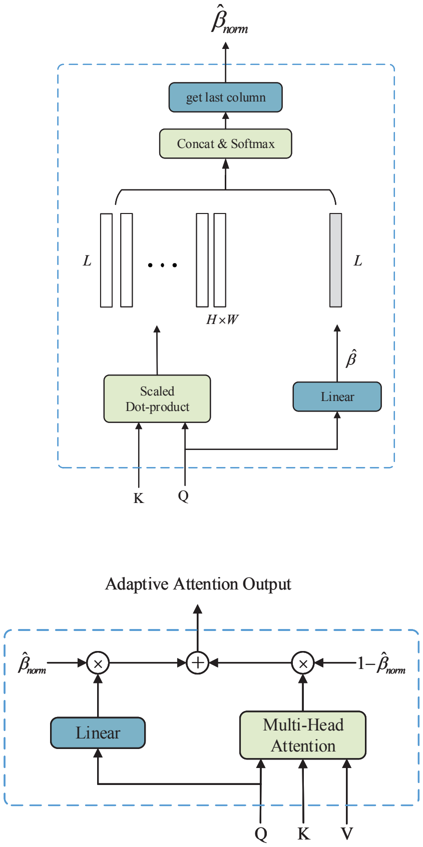


Fig. 2: (top) Module for calculating adaptive gate. (down) Adaptive Multi-Head Attention module.

Caption generation is a process of language modeling. The image feature and generated word are put into the decoder at each timestep. The decoder outputs a probability vector P_{vocab}, where *vocab* indicates the dictionary size of the current dataset. In general select the index which corresponds to highest probability, take the word in this index as the output at current timestep. In order to make probability higher in the index with respect to ground-truth, and the ground-truth is one-hot data, so the optimization goal is to maximize the logarithm sum of P_{vocab}, or minimize the cross entropy loss, as follows:

$$\theta^* = \arg\min_{\theta} \sum_{(V,y)} -\log p\left(y|V';\theta\right) \tag{1}$$

Where θ is parameter of the decoder, V' is spatial feature vector, probability vector is $P_{vocab} = \{p_1, \ldots, p_L\}$, $p_i \in R^{vocab}$, L is the length of the sentence. Sample in P_{vocab} to obtain the word vector $y = \{y_1, y_2, \ldots, y_L\}$, y_i represents *i-th* word.

B. Adaptive Attention in Transformer

1) Scaled Dot-Product Spatial Attention: During the process of generating captions, different regions of the image contribute to the decoder in different degree, so it is necessary to use spatial attention to dynamically assign different weight to each region. Suppose the state of the decoder is query Q. We use key-value pairs to represent the output of the encoder, K and V are equal and both the features of the image. Firstly, calculate the correlation between Q and K to get a weight vector. Secondly, use this vector to weight V, the operation is as shown in the following equation:

$$V_{\text{spatial\_weight}} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

Where $Q \in R^{L\times d_k}$, L is the length of the sentence, d_k is the feature dimension corresponding to each head in the multi-head attention model, for details about multi-head, see section 3). $K, V \in R^{(H\times W)\times d_k}$, where H and W are the height and width of the spatial feature map respectively. Here we use dot-product to calculate the correlation between Q and K. In order to ensure that the calculation result is within the valid range, use $\sqrt{d_k}$ to scale it. Then employ a softmax function to obtain a probability distribution, finally perform the dot-product with V to get the output.

Note the above attention mechanism is between the encoder and decoder, which is corresponding to the second sub-layer of the Transformer. There is also a self-attention mechanism in the model, which is in the first sub-layer. This mechanism attends to the correlation within sentence. The calculation process is same as the above, but only use equal Transformer intermediate state for K, V and Q.

2) Adaptive Attention: In spatial attention based approach image feature is also input to decoder at every timestep. In order to ensure the image feature input to decoder only when it is needed. We employ the adaptive gate same as visual sentinel [9] to determine when to use the image feature. When it comes to some words which have low correlation to image, the adaptive gate is able to depend on Transformer entirely. If only use Transformer, the model could infer next word from generated sentence, otherwise the image feature will influence the output of Transformer. The adaptive attention module is shown in Fig. 2.

The decoder consists of multiple Transformer layers. The image feature is input to the decoder at the first layer. Therefore latter layer state of the decoder contains visual information as well as linguistic information. Here we introduce a parameter $\beta \in R^{d_k\times 1}$ to extract state information. Calculate the dot-product of Q and β to obtain $\hat{\beta} = Q\beta$, where $\hat{\beta} \in R^{L\times 1}$, this vector contains the decoder's state or memory, based on this vector we calculate the adaptive gate as following:

$$\alpha = \mathrm{softmax}([\frac{QK^T}{\sqrt{d_k}};\hat{\beta}]) \tag{3}$$

Where $QK^T \in R^{L\times(H\times W)}$ is the dot-product of the decoder state Q and the image spatial feature K. Every row represents a weight distribution of the image regions, when decoder generates the *i-th* word. Concatenate this vector with $\hat{\beta}$ in column direction, then normalize result to probability with softmax operation. At last we select the last column of the matrix to get $\hat{\beta}_{norm} = \alpha[H \times W + 1]$, where $\hat{\beta}_{norm} \in [0, 1]$, note $\hat{\beta}_{norm}$ is adaptive gate vector, the value of this vector can determine whether to use image information. Calculation process is responding to the top part of Fig. 2.

$$V_{\mathrm{adap\_weight}} = \hat{\beta}_{\mathrm{norm}} \odot QW_m + \left(1 - \hat{\beta}_{\mathrm{norm}}\right) \odot V_{\mathrm{spatial\_weight}} \tag{4}$$

Use the above method to weight average the encoder's output, this calculation process is responding to the below part of Fig. 2. Where $V_{\mathrm{adap\_weight}}$ is the output of the adaptive attention, $QW_m, V_{\mathrm{spatial\_weight}} \in R^{L\times d_k}$. $\odot$ represents element-wise product, $W_m \in R^{d_k\times d_k}$ is a trainable parameter, QW_m vector contains the state information of the decoder, $V_{\mathrm{spatial\_weight}}$ represents the spatial information of the image. When $\hat{\beta}_{\mathrm{norm}} = 0$ decoder uses image information entirely, while $\hat{\beta}_{\mathrm{norm}} = 1$ indicates decoder only uses its previous information. So adaptive vector guarantees the Transformer's input is adaptive when generating the sequence.

3) Multi-Head Attention: In convolutional neural network, multiple convolution kernels are used to extract different features. Inspired by this idea, Transformer uses multi-head attention to make feature become diversity.

Set $d_{model} = h * d_k$, d_k is the dimension of each single-head, h is the number of multi-head, where d_{model} represents dimension of input and intermediate state. In experiment the dimension of the input image is $R^{(H\times W)\times d_{model}}$, the dimension of the caption is $R^{L\times d_{model}}$. Decompose the input vector into h parts to achieve multi-head, then each single-head part performs adaptive attention respectively. Here use function AdaptiveAttention() to represent the operation in (4). Finally concatenate h outputs of single-head, use a linear to map the outputs to obtain the eventual value as following.

$$\mathrm{head}_i = \mathrm{AdaptiveAttention}\left(Q_i, K_i, V_i\right) \tag{5}$$

$$\mathrm{MultiHead} = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W_o \tag{6}$$

There are two types of multi-head attention in Transformer: Masked self-attention and Encoder-Decoder attention. The first sub-layer of Transformer is the first attention, in which Q_i, K_i and V_i are equal, they are all the intermediate state vector of Transformer. Their dimensions are all $R^{L\times d_k}$, in this part there is a mask to ensure the flow of information according to auto-regressive property. Second sub-layer of Transformer belongs to the second attention, where Q_i, K_i and V_i are responding to (3), so $Q \in R^{L\times d_k}$ and $K, V \in R^{(H\times W)\times d_k}$, $W_o \in R^{(h\times d_k)\times d_{model}}$.

C. Training and Inference

There is sequence dependency inside LSTM, whether in training or inference step LSTM only generates one word at every timestep. While Transformer is entirely dependent on attention mechanism, it uses positional encoding to memory the order of the sequence, during the training step, the label caption embedding matrix $S \in R^{L\times d_{model}}$ and spatial image matrix $V' \in R^{(H\times W)\times d_k}$ are known, thus Transformer can output probability matrix for the whole sentence in the meantime with Teacher Force method. Therefore train Transformer is more efficient.

During inference step, Transformer is same as LSTM. They both need previous output $p_{t-1} \in R^{d_{vocab}}$ as input in current time. We can use greedy search to select the index with the highest probability, or use beam search to find the combination of word sequences which has largest probability totally.

IV. Experiments

A. Datasets and settings

We test our model on Flickr30k [16] dataset, in which the sample is image-captions pairs. Every image has 5 captions,

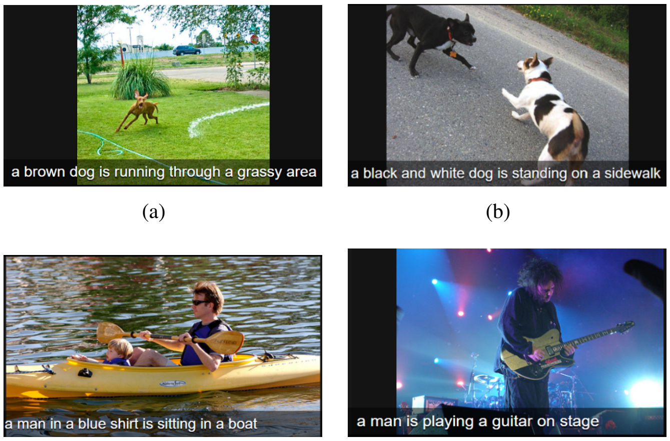
TABLE I: Performance on Flickr30k compared to other methods, (-) indicates an unknown metric.

	Flickr30k					
Method	***B-1***	***B-2***	***B-3***	***B-4***	***METEOR***	***CIDER***
NIC [2]	0.663	0.423	0.277	0.183	-	-
DeepVS [3]	0.573	0.369	0.240	0.157	-	0.247
Soft-Attention [8]	0.667	0.434	0.288	0.191	0.185	-
Adaptive-LSTM [9]	0.667	0.494	0.354	0.251	0.204	**0.531**
Ours	**0.670**	**0.496**	**0.355**	**0.252**	**0.204**	0.530

and most describe human activities, the style of captions are relatively fixed, such as ”a football player is running.” We use the split method in [3]. The training set includes 29000 samples, the validation set and the test set both have 1000 samples. We perform a random data augmentation in the training step. Use CNN ResNet-101 as encoder, the input size of the image is 256×256, and set the adaptive pooling after the last convolution be 7. The output of the image is $(7, 7, 2048)$, here 2048 is the number of channels of the feature map. The Transformer hyperparameters use the settings in [6], input dimension is $d_{model} = 512$, internal feedforward layer's dimension is $d_{ff} = 2048$ and the number of the model layers is $N = 6$. Finally use the Adam optimizer with warm up approach to optimize the Transformer.

B. Performance on Flickr30k

BLEU metric was originally used to evaluate the machine translation task, this metric uses the precision of the n-gram between the generated and the ground-truth sentence, usually four n-gram metric from BLEU-1 to BLEU-4 are commonly used. METEOR metric uses the explicit word match method to compare generated sentence with ground-truth sentence. All of these metrics are used in NLP tasks. But CIDER metric is designed for image caption task specifically, it evaluates the consensus between generated sentence and ground-truth sentence. Here we use these 6 metrics to evaluate the performance, then compare it with other models on the same dataset. The results are shown in Table 1.


a brown dog is running through a grassy area
a black and white dog is standing on a sidewalk
a man in a blue shirt is sitting in a boat
a man is playing a guitar on stage

(a)

(b)

(c)

(d)

Fig. 3: The caption results in test set.

The last row in Table 1 is our model. It can be seen that most metrics are improved compared to other models, so our Adaptive-Trans model can really improve performance. Compared to original adaptive model based on LSTM, our model outperforms it slightly in most metric but CIDER, this due to the decoder's improvement, in which Transformer is more powerful than LSTM. We random select some samples from the test set and generate captions using Adaptive-Trans model. The results are shown in Fig. 3.

V. Conclusion

In our work, we combine the spatial and adaptive attention into Transformer, the model learns to determine where and when to use the image feature, this method makes the generated caption more accurate. Not only the model can improve the original performance, but also boost the training speed. In future work, more powerful encoder and decoder will substitute the old structure, and more advanced modules will introduce to the caption model to improve the performance and efficiency. Of course more methods will be proposed to make the generated caption more close to human level.

Acknowledgment

This work was supported by National Natural Science Foundation of China under grants 61520106009, 61533008 and 61703037.

References

[1] J. Mao, W. Xu, Y. Yang, et al, “Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN),” International Conference on Learning Representations, 2015.

[2] O. Vinyals, A. Toshev, S. Bengio, et al, “Show and tell: A neural image caption generator,” IEEE Conference on Computer Vision and Pattern Recognition, 3156-3164, 2015.

[3] A. Karpathy, F. Li, “Deep visual-semantic alignments for generating image descriptions,” IEEE Conference on Computer Vision and Pattern Recognition, 3128-3137, 2015.

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, et al, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” arXiv preprint arXiv:1406.1078, 2014, 2014.

[5] K. He, X. Zhang, S. Ren, et al, “Deep residual learning for image recognition,” IEEE Conference on Computer Vision and Pattern Recognition, 770-778, 2016.

[6] A. Vaswani, N. Shazeer, N. Parmar, et al, “Attention is all you need,” Advances in neural information processing systems, 5998-6008, 2017.

[7] D. Bahdanau, K. Cho, Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” Computer Science, 2014.

[8] K. Xu, J. Ba, R. Kiros, et al, “Show, attend and tell: Neural image caption generation with visual attention,” International conference on machine learning, 2048-2057, 2015.

[9] J. Lu, C. Xiong, D. Parikh, et al, “Knowing when to look: Adaptive attention via a visual sentinel for image captioning,” IEEE Conference on Computer Vision and Pattern Recognition, 375-383, 2017.

[10] M. Hodosh, P. Young, J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *Journal of Artificial Intelligence Research*, 47: 853-899, 2013.

[11] A. Farhadi, M. Hejrati, M. A. Sadeghi, et al, "Every picture tells a story: Generating sentences from images," European conference on computer vision, 15-29, 2010.

[12] Q. You, H. Jin, Z. Wang, et al, "Image Captioning with Semantic Attention," IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[13] Q. Wu, C. Shen, L. Liu, et al, "What value do explicit high level concepts have in vision to language problems?," IEEE Conference on Computer Vision and Pattern Recognition, 203-212, 2016.

[14] J. Johnson, A. Karpathy, F. Li, "Densecap: Fully convolutional localization networks for dense captioning," IEEE Conference on Computer Vision and Pattern Recognition, 4565-4574, 2016.

[15] S. J. Rennie, E. Marcheret, Y. Mroueh, et al, "Self-critical sequence training for image captioning," IEEE Conference on Computer Vision and Pattern Recognition, 7008-7024, 2017.

[16] P. Young, A. Lai, M. Hodosh, et al, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," Transactions of the Association for Computational Linguistics, 2: 67-78, 2014.