Generating Video Description using RNN with Semantic Attention

Natsuda Laokulrat[1], Naoaki Okazaki[2,1] and Hideki Nakayama[3,1]

[1]AIRC, AIST
[2]Tohoku University
[3]The University of Tokyo

Abstract

Being able to understand videos can have a great impact and can be useful to many other applications. However, generated descriptions by computers often fail to mention correct objects appearing in videos. This work aims to alleviate this problem by including external fine-grained visual information detected from all video frames. In this paper, we propose an LSTM-based sequence-to-sequence model with semantic attention for video description generation. The results show that using semantic attention to selectively focus on external fine-grained visual information can guide the system to correctly mention objects in videos and have a better quality of video descriptions.

1 Introduction

Automatic video description generation has been tackled by the combination of RNN and CNN. Venugopalan et al. (2015b) proposed the first end-to-end system to translate a video into natural language by extending the CNN-RNN encoder-decoder framework for image captioning proposed by Vinyals et al. (2014) to generate descriptions for videos. They performed a mean pooling over CNN feature vectors of frames to generate a single vector representation for a video, and then use the vector as input to the RNN decoder to generate a sentence.

Later, they have proposed an RNN-based sequence-to-sequence model for generating descriptions of videos (Venugopalan et al., 2015a). They used 2 layers of RNN for both encoding the videos and decoding into sentences, so their model is able to learn both a temporal structure of a sequence of video frames and a sequence model for generating sentences.

However, one problem of video description generation is that generated descriptions by computers often fail to mention correct objects and actions appearing in videos. Inspired by the image captioning model with semantic attention proposed by You et al. (2016), in this paper, we present a sequence-to-sequence encoder-decoder model with semantic attention mechanism, which is a novel approach to integrate fine-grained visual information appearing in video frames to help the model generate descriptions. The results show that the semantic attention mechanism can guide the system to correctly mention objects and actions, and a have better quality of video descriptions.

2 LSTM encoder-decoder model

Given an input x_t, at time step t, one unit of an LSTM can be formulated as

$$\begin{aligned} i_t &= sigmoid(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= sigmoid(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ o_t &= sigmoid(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ g_t &= tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \quad (1)$$

where i_t, f_t and o_t are input gates, forget gates, and output gates. The symbol $\odot$ represents the element-wise multiplication. W_{xi}, W_{hi}, W_{xf}, W_{hf}, W_{xo}, W_{ho}, W_{xg}, W_{hg} and b_i, b_f, b_o, b_g are the parameters to be learned during training. h_t is the hidden state at time step t which will be an input to the next time step's LSTM unit.

2.1 Non-attention model

Figure 1 depicts our two-layer LSTM model for generating description sentence from a video. Given a video as a sequence of frames $V = \{v_1, v_2, ..., v_n\}$ where the video V has n frames and v_t is the t^{th} frame of the video. The input

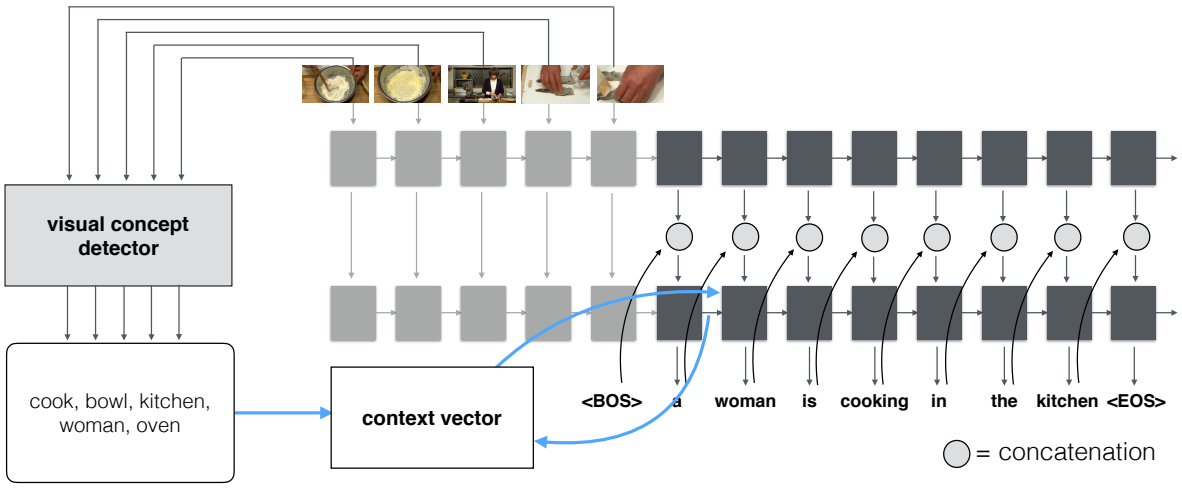

Figure 1: System architecture of our model. In the figure, we omit the image embedding layer, the word embedding layer, and the softmax layer, due to the space constraint.

frames x_t can be described as

$$x_t = \begin{cases} v_t & , t \leq n \\ \vec{0} & , t > n \end{cases} \quad (2)$$

The video frames are taken as input one by one at encoding time, and are set to $\vec{0}$ at decoding time. Then, we can formulate our the first (upper) LSTM layer as

$$h_t^{(1)} = LSTM^{(1)}(x_t, h_{t-1}^{(1)}) \quad (3)$$

where $h_t^{(1)}$ is the hidden state of the first LSTM layer, defined as $LSTM^{(1)}$, at time step t.

The input to the second (lower) LSTM layer is the concatenation of the word generated on the previous time step w_{t-1} and the hidden state of the first LSTM layer.

$$h_t^{(2)} = LSTM^{(2)}([w_{t-1}; h_t^{(1)}], h_{t-1}^{(2)}) \quad (4)$$

where $h_t^{(2)}$ is the hidden state of the second LSTM layer, defined as $LSTM^{(2)}$, at time step t.

At encoding time, w_{t-1} is set to $\vec{0}$ since there is no word being generated. The distribution over all the words at time step t can be computed by

$$p(w_t|w_1, ..., w_{t-1}, V) = softmax(W_s h_t^{(2)} + b_s) \quad (5)$$

As with Venugopalan et al. (2015a), we have $x_t = \vec{0}$ at encoding time and $w_{t-1} = \vec{0}$ at decoding time in order to use a single LSTM (for one layer) that learns both encoding and decoding, so the weights can be shared.

2.2 Semantic attention model

Given a set of visual concepts of the video $S = \{s_1, s_2, ..., s_k\}$ where s_i can be represented by a word vector in the same space as word input. The second-layer LSTM at decoding time can be formulated as

$$h_t^{(2)} = LSTM^{(2)}([w_{t-1}; c_t; h_t^{(1)}], h_{t-1}^{(2)}) \quad (6)$$

where the context vector c_t, at the time step t in the decoding stage, is the weighted sum of visual concepts.

$$c_t = \sum_{i=1}^{k} a_t(i) s_i \quad (7)$$

The weight $a_t(i)$ is computed at every time step t by

$$a_t(i) = \frac{e^{score(h_{t-1}^{(2)}, s_i)}}{\sum_{j=1}^{k} e^{score(h_{t-1}^{(2)}, s_j)}} \quad (8)$$

where $score(h_{t-1}^{(2)}, s_i)$ is the score function used to calculate alignment weights between every visual concept s_i and the hidden state $h_{t-1}^{(2)}$.

$$score(h_{t-1}^{(2)}, s_i) = v_a^\top tanh(W_a[h_{t-1}^{(2)}; s_i]) \quad (9)$$

The parameters W_a and v_a of the score function are jointly learned during training.

3 Experiment

3.1 Dataset and pre-processing

We use *Microsoft Research Video Description Corpus (MSVD)* (Chen and Dolan, 2011) which is a set of 1,970 Youtube clips with

Model	BLEU	METEOR	CIDEr	ROUGE-L
Results reported by Venugopalan et al. (2015a)				
Mean pooling (VGG16)	-	0.277	-	-
Sequence to sequence (VGG16)	-	0.292	-	-
Sequence to sequence (VGG16) + Flow (AlexNet)	-	**0.298**	-	-
Our system (VGG16) - *non-attention*	**0.393**	0.308	0.580	**0.666**
Our system (VGG16) - *semantic attention*	0.386	**0.313**	**0.589**	0.665

Table 1: Scores of video description generation results on the MSVD dataset.

≈40 captions/clip. We split the dataset into train/validation/test sets following Venugopalan et al. (2015b).

We downsample the video clips by selecting every 8^{th} frame and resize them to 224x224. Then, we extract features for each frame using a pre-trained image classification model provided in *Caffe Model Zoo* (Jia et al., 2014). In this work, we use the 4096-dimensional fc7 layer of the VGG16 model (Simonyan and Zisserman, 2014) as frame features and embed them into 512-dimensional embeddings.

For text input, we represent words with GloVe pre-trained word embeddings, proposed by Pennington et al. (2014). We map the 300-dimensional GloVe word vectors into 1000-dimensional vectors. The visual concepts are treated in the same way as text input.

3.2 Experiment setting

In order to enable batch training, we constrain the number of encoding and decoding time steps to be 60 and 20, respectively. We use the Adam optimizer with the learning rate of 0.0001 and the mini-batch size of 200. The LSTM hidden layer size is set to 1,000. To avoid overfitting, we apply the dropout strategy with the ratio of 0.3 at the frame input layer. All the parameters are jointly learned at training time.

3.3 Visual concept detection

We use the pre-trained model provided by Fang et al. (2015) to detect visual concepts from every frame of the downsampled videos. The visual concepts includes actions, objects, attributes of objects, and also locations.

The detected visual concepts of all frames of a video are combined into one collection. For one video, we select 20 concepts from the collection and treat them equally, ignoring their scores provided by the concept detector, as shown in the boxes in Figure 2.

3.4 Evaluation

We performed a quantitative analysis of results based on four evaluation metrics, including BLEU, METEOR, CIDEr, and ROUGE-L.

We implemented our system using *Chainer* (Tokui et al., 2015) and used the caption evaluation package provided by the Microsoft COCO Image Captioning Challenge (Chen et al., 2015).

3.5 Experimental results

Table 1 shows the experimental results of our proposed system. METEOR and CIDEr scores slightly increased while BLEU and ROUGE-L scores dropped when using semantic attention.

Even though the semantic attention mechanism cannot clearly improve the scores of the test set, we can see some promising results in Figure 2. The relevant visual concepts were focused and the alignment weights changed properly when each word of the sentences were being generated.

In the bottom-right example, though the model mistakenly focused on the visual concept *'woman'*, the mis-mentioned object *(bicycle)* in non-attention model can be correctly identified *(bike)* by the model with semantic attention.

We can see that many irrelevant visual concepts were detected. This is because we used the visual concept detector that was trained on other datasets, so it could not perform well in MSVD video frames. We believe that if we re-train the concept detector with our dataset, we can achieve better results.

4 Conclusion

In this paper, we have proposed an LSTM-based sequence-to-sequence model with semantic attention for video description generation. The scores do not have obvious improvement; however, the model is able to learn to focus on external fine-grained information of videos and a have better quality of video descriptions.

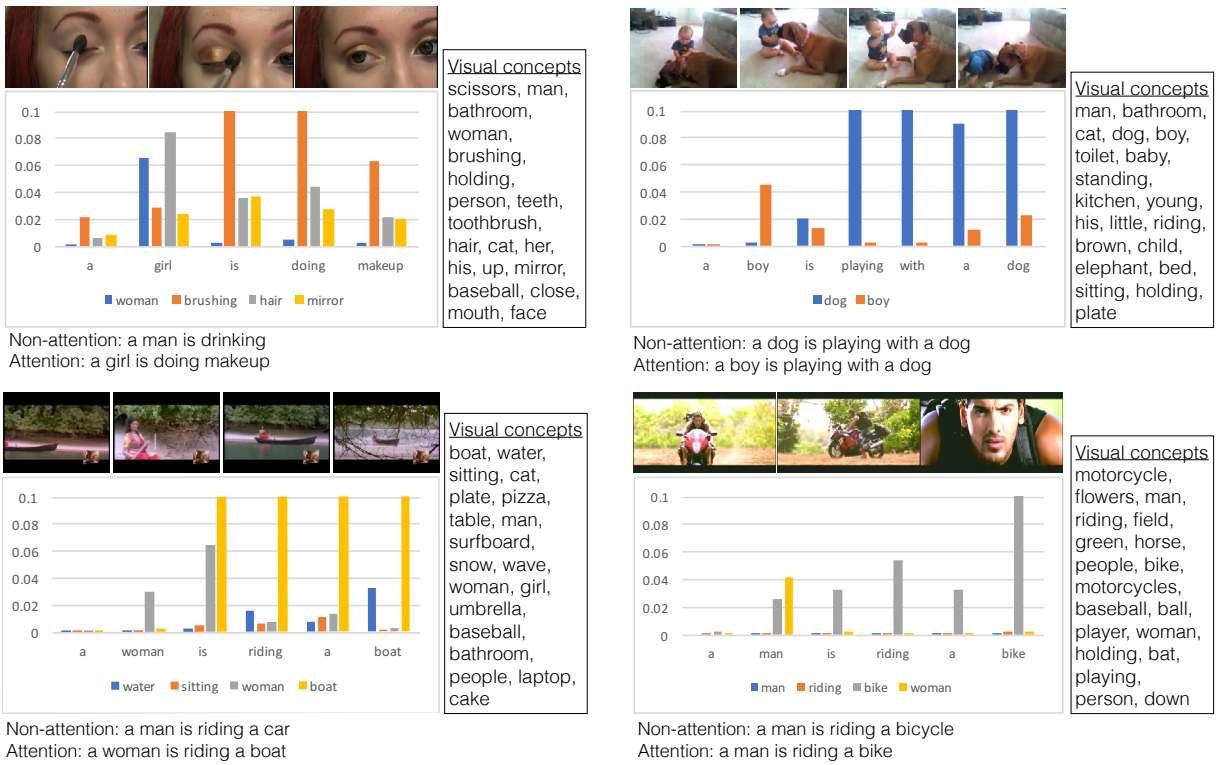

Figure 2: Example of generated descriptions and alignment weights of visual concepts when each word of the sentences was generated. The values are clipped at 0.1 for easier reading.

Acknowledgement

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

David L. Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *ACL 2011*.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv:1504.00325*.

Hao Fang, Saurabh Gupta, Forrest N. Iandola, Rupesh Kumar Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C. Platt, C. Lawrence Zitnick, and Geoffrey Zweig. 2015. From captions to visual concepts and back. In *CVPR 2015*.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP 2014*.

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

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015a. Sequence to sequence – video to text. In *ICCV 2015*.

Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2015b. Translating videos to natural language using deep recurrent neural networks. In *NAACL-HLT 2015*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *arXiv:1411.4555*.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *CVPR 2016*.