Incorporating Semantic Attention in Video Description Generation

Natsuda Laokulrat[1], **Naoaki Okazaki**[1,2], **Hideki Nakayama**[1,3]
[1]Artificial Intelligence Research Center (AIRC), AIST, Japan
[2]Tokyo Institute of Technology, Japan
[3]The University of Tokyo, Japan
natsudalkr@gmail.com, okazaki@c.titech.ac.jp, nakayama@ci.i.u-tokyo.ac.jp

Abstract

Automatically generating video description is one of the approaches to enable computers to deeply understand videos, which can have a great impact and can be useful to many other applications. However, generated descriptions by computers often fail to correctly mention objects and actions appearing in the videos. This work aims to alleviate this problem by including external fine-grained visual information, which can be detected from all video frames, in the description generation model. In this paper, we propose an LSTM-based sequence-to-sequence model with semantic attention mechanism for video description generation. The model is flexible so that we can change the source of the external information without affecting the encoding and decoding parts of the model. The results show that using semantic attention to selectively focus on external fine-grained visual information can guide the system to correctly mention objects and actions in videos and have a better quality of video descriptions.

Keywords: video description generation, deep learning, RNN, attention

1. Introduction

In the past few years, the image captioning task has been gaining popularity among researchers and high-quality image captions can be generated by deep learning techniques (Vinyals et al., 2014; Karpathy and Fei-Fei, 2015; Fang et al., 2015; Xu et al., 2015; You et al., 2016). The first work in this field was proposed by Vinyals et al. (2014). They proposed an end-to-end system consisting of a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN). The output of the last fully connected layer of the image classification CNN is used as an image feature and then is injected into the RNN-based language model to produce a meaningful sentence. Later, Xu et al. (2015) has proposed an attention-based framework for image captioning which can selectively focus on a portion of an image while producing each word. However, researchers still cannot achieve a satisfying quality of video descriptions generated by computers yet. As with image captioning, automatic video description generation combines two fields of artificial intelligence, computer vision and natural language processing, and has also been tackled by the combination of RNN and CNN.

Venugopalan et al. (2015b) proposed the first end-to-end system to translate a video to 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, ignoring the temporal ordering of videos. Subsequently, 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 to 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. Later, Laokulrat et al. (2016) applied the temporal attention mechanism to the sequence-to-sequence model to focus on a set of frames while generating each word of the describing sentence. With their attention mechanism, they were able to improve the scores without using any additional features. Yao et al. (2015) used CNN for encoding video frames and used RNN for building a language model at decoding time. They have also incorporated an attentional mechanism to video caption generation, taking into account both local and global temporal structures of videos by incorporating a spatial temporal 3D CNN.

Venugopalan et al. (2017) have attempted to describe objects unseen in paired image-text training data, by taking advantage of other external sources, e.g. labeled images from object recognition datasets, and semantic knowledge extracted from unannotated text.

One problem of video description generation is that generated descriptions by computers often fail to mention correct objects and actions appearing in videos. With the previously proposed sequence-to-sequence model, the video showed in Figure 1 is described as *'a man is riding a car'*. It is obvious that the model cannot detect the main subject *'woman'* and the object *'boat'* appearing in the video. In this work, we aim to solve this problem by integrating external information seamlessly to the conventional sequence-to-sequence model. We want the model to be flexible enough so that we can change the source of the external information without affecting the encoding and decoding parts of the model. 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. By performing a set of experiments, the results show that the semantic atten-

This work has been done while the first author was working at AIRC, AIST.

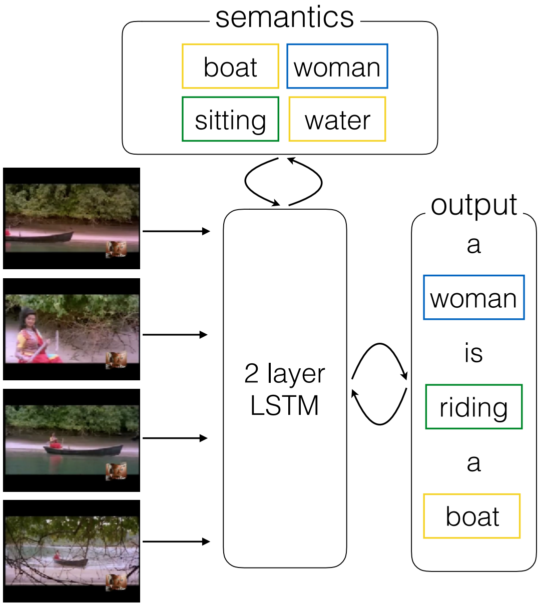


Figure 1: Using semantic attention in video description generation. The system attends to particular semantic words while generating each word of the output sentence. It focuses on the semantic *'woman'* while producing the word *'woman'* (see blue boxes) and focuses on the semantic words *'boat'* and *'water'* while producing the word *'boat'* in the output sentence (see yellow boxes).

tion mechanism can guide the system to correctly mention objects and actions, and a have better quality of video descriptions.

In Figure 1, the semantic-attention model attends to (a) the semantic word *'woman'* when producing with output word *'woman'* (see blue boxes), (b) the semantic word *'sitting'* when producing with output word *'riding'* (see green boxes), (c) the semantic words *'boat'* and *'water'* when producing the output word *'boat'* (see yellow boxes). By integrating the external information, we can fix the error words *'man'* and *'car'* in the output sentence and recover the correct objects *'woman'* and *'boat'*.

The rest of this paper is organized as follows: Section 2 describes in detail the non-attention model and and Section 3 introduces the semantic attention model, along with the mathematical formulas. Section 4 shows the experiments and the results in both qualitative and quantitive aspects. Section 5 discusses the experimental results and Section 6 concludes this paper.

2. LSTM encoder-decoder model

This section explains a sequence-to-sequence model (without attention mechanism) for generating video description. The model is based on the work previously proposed by Venugopalan et al. (2015a).

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. We extract a frame feature $e_t \in \mathbb{R}^{d_e}$ of each frame v_t by using a pre-trained image classification model IM, where d_e is the dimensionality of the original frame feature. The input feature e_t of the input frame v_t can be described as

$$e_t = \begin{cases} IM(v_t) & ,t \le n \\ \vec{0} & ,t > n \end{cases} \tag{1}$$

Then, we embed it into a lower-dimensional vector $x_t = W_{ex}e_t \in \mathbb{R}^{d_x}$, where d_x is the dimensionality of the embedded frame feature and $W_{ex} \in \mathbb{R}^{d_x \times d_e}$ is a weight matrix. We omit bias terms for simplicity.

As depicted in the light grey area in Figure 2, the video frames are taken as input to the LSTMs one by one at encoding time, and are set to $\vec{0}$ at decoding time. Then, we can formulate the first (upper) LSTM layer as

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

where $h_t^{(1)} \in \mathbb{R}^{d_h}$ is the hidden state of the first LSTM layer, defined as $LSTM^{(1)}$, at time step t, and d_h is the dimensionality of the hidden states. The hidden state is initialized with a zero vector.

Let $w_t \in \mathbb{R}^r$, where r is the vocabulary size including $\langle UNK \rangle$, $\langle BOS \rangle$ and $\langle EOS \rangle$, be the word generated at time t. The input to the second (lower) LSTM layer is the concatenation of the hidden state of the first LSTM layer and the embedding of the word generated on the previous time step $q(w_{t-1}) \in \mathbb{R}^{d_q}$, where d_q is the dimensionality of the word embedding. So, the second LSTM layer can be described as

$$h_t^{(2)} = LSTM^{(2)}([q(w_{t-1}); h_t^{(1)}], h_{t-1}^{(2)}) \tag{3}$$

where $h_t^{(2)} \in \mathbb{R}^{d_h}$ is the hidden state of the second LSTM layer, defined as $LSTM^{(2)}$, at time step t. Note that the dimensionality of $h_t^{(1)}$ and $h_t^{(2)}$ is the same. At encoding time, w_{t-1} is set to $\vec{0}$ since there is no actual word being generated.

Let $Y = \{y_1, y_2, ..., y_m\} \in \mathbb{R}^r$ be the target sentence with m words. Lastly, the distribution over all the words at time step t can be computed by taking softmax over all possible words. This can be formulated as

$$p(w_t | w_1, ..., w_{t-1}, V) = softmax(W_s h_t^{(2)}) \tag{4}$$

where $W_s \in \mathbb{R}^{r \times d_h}$ is a weight matrix. The decoding process iterates until $\langle EOS \rangle$ symbol is produced. The model can be trained end-to-end by minimizing the softmax cross-entropy loss between y_t and w_t. The loss is computed only in decoding time.

At training time, we use y_{t-1} as an input to Equation 3 instead of w_{t-1}. 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 share the weights of the encoding and decoding LSTMs. This can help speed up the model training without losing much accuracy.

3. Semantic attention model

In this work, *'semantic'* refers to any visual concepts appearing in the video frames, including object, actions, color, shape, relationship, and so on. The objective of incorporating semantic attention mechanism is to enable the language model to focus on related concepts when producing each word of a sentence.

Figure 2 depicts our two-layer LSTM model with semantic attention for generating description sentence from a

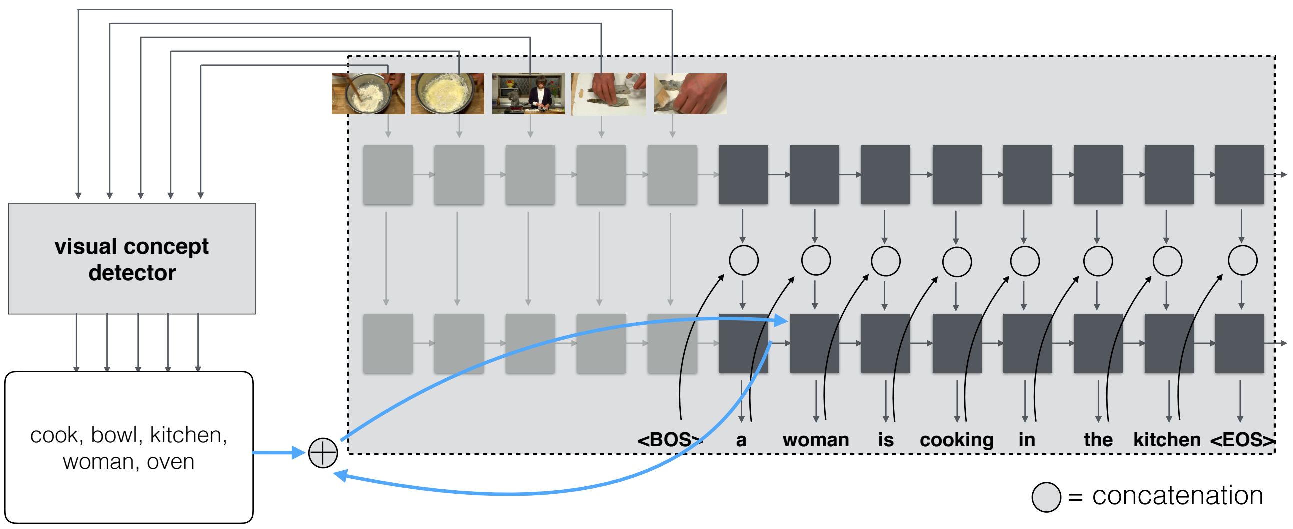


Figure 2: System architecture of the sequence-to-sequence model with semantic attention. In the figure, we omit the image embedding layer, the word embedding layer, and the softmax layer, due to the space constraint. The light grey area shows the non-attention model.

video. Given a set of visual concepts of the video $S = \{s_1, s_2, ..., s_k\} \in \mathbb{R}^{d_q}$ where s_i can be represented by a word embedding in the same space as word input $q(w_{t-1})$. In the semantic attention model, the second-layer LSTM at decoding time can be formulated as

$$h_t^{(2)} = LSTM^{(2)}([q(w_{t-1}); c_t; h_t^{(1)}], h_{t-1}^{(2)}) \quad (5)$$

where the context vector $c_t \in \mathbb{R}^{d_q}$, at the time step t in the decoding stage, is the weighted sum of visual concepts. The context vector c_t can be calculated by

$$c_t = \sum_{i=1}^{k} a_t(i) s_i \quad (6)$$

where $a_t(i) \in \mathbb{R}^1$ is the alignment weight of semantic word (visual concept) s_i at time step t. The weight $a_t(i)$ is computed at every decoding time step t by

$$a_t(i) = \frac{e^{score(h_{t-1}^{(2)}, s_i)}}{\sum_{j=1}^{k} e^{score(h_{t-1}^{(2)}, s_j)}} \quad (7)$$

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 $h_{t-1}^{(2)}$. The score function can be formulated as

$$score(h_{t-1}^{(2)}, s_i) = v_a^\top \cdot tanh(W_a[h_{t-1}^{(2)}; s_i]) \quad (8)$$

The parameters $W_a \in \mathbb{R}^{(d_h+d_q)\times(d_h+d_q)}$ and $v_a \in \mathbb{R}^{(d_h+d_q)}$ of the score function are jointly learned during training.

3.1. 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 include 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 5.

Sentence	Number of occurrences
a man is playing a guitar	217
a man cooking his kichen	196
a man is playing guitar	115
a woman is riding a horse	86
a man is playing the guitar	74
a baby is laughing	67
a person is cooking	59
a cat is playing	57
a woman is peeling a potato	53
a man is singing	52

Table 1: 10 most frequently occurring sentences in the training set and the number of occurrences of each sentence.

4. Experiment

This section explains the dataset we used, the pre-processing steps we performed, the visual concept detection process, the experiment setting as well as the experimental results.

4.1. 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. For fair comparison with previous work, we split the dataset into train/validation/test sets following (Venugopalan et al., 2015b) and (Yao et al., 2015). The size of the train, validation, and test sets is 1,200, 100, and 670, respectively.

Figure 3 shows the histogram of the number of captions per a video clip in the training set. The average number is ≈40 captions/clip. The minimum number of sentence per a video clip is 18, and the maximum is 66.

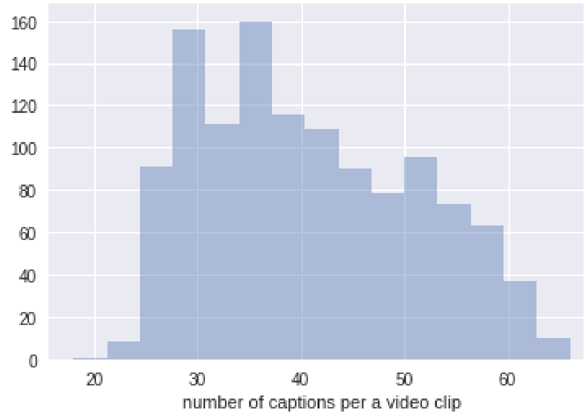


Figure 3: Histogram of number of captions per a video. mean = 40.65, min = 18, max = 66.

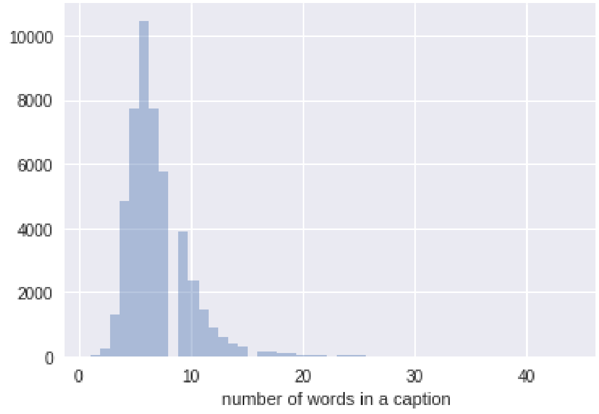


Figure 4: Histogram of number of words in a caption. mean = 7.03, min = 1, max = 45.

Most of the captions contain a single activity and can be described using only one sentence as shown in Table 1. Table 1 shows 10 most frequently occurring captions in the training set. The sentence *'a man is playing a guitar'* appears 217 times which is the maximum number in the training data.

Figure 4 shows the histogram of the number of words per a caption. The average length of the captions in the training set is ≈7 words. The minimum and maximum length are 1 and 45 words respectively.

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, the pre-processing includes tokenizing, converting to lower case, and removing punctuations. We represent words with GloVe pre-trained word embeddings, proposed by Pennington et al. (2014). The words outside pre-trained GloVe embeddings are converted to $\langle UNK \rangle$. We then map the 300-dimensional GloVe word vectors into 1000-dimensional vectors. The visual concepts are treated in the same way as text input.

4.2. 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 (Kingma and Ba, 2014) 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 (Srivastava et al., 2014) with the ratio of 0.3 at the frame input layer. All the parameters are jointly learned at training time. We apply the beam search strategy at decoding time (beam size = 5).

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). We performed a quantitative analysis of results based on four evaluation metrics, including

BLEU (Papineni et al., 2002), a precision-based evaluation metric used in machine translation.

METEOR (Denkowski and Lavie, 2014), an automatic metric for machine translation evaluation based on explicit word-to-word matching.

CIDEr (Vedantam et al., 2014), an automatic consensus metric of image description quality.

ROUGE-L (Lin, 2004), a recall-oriented evaluation metric popularly used in summarization.

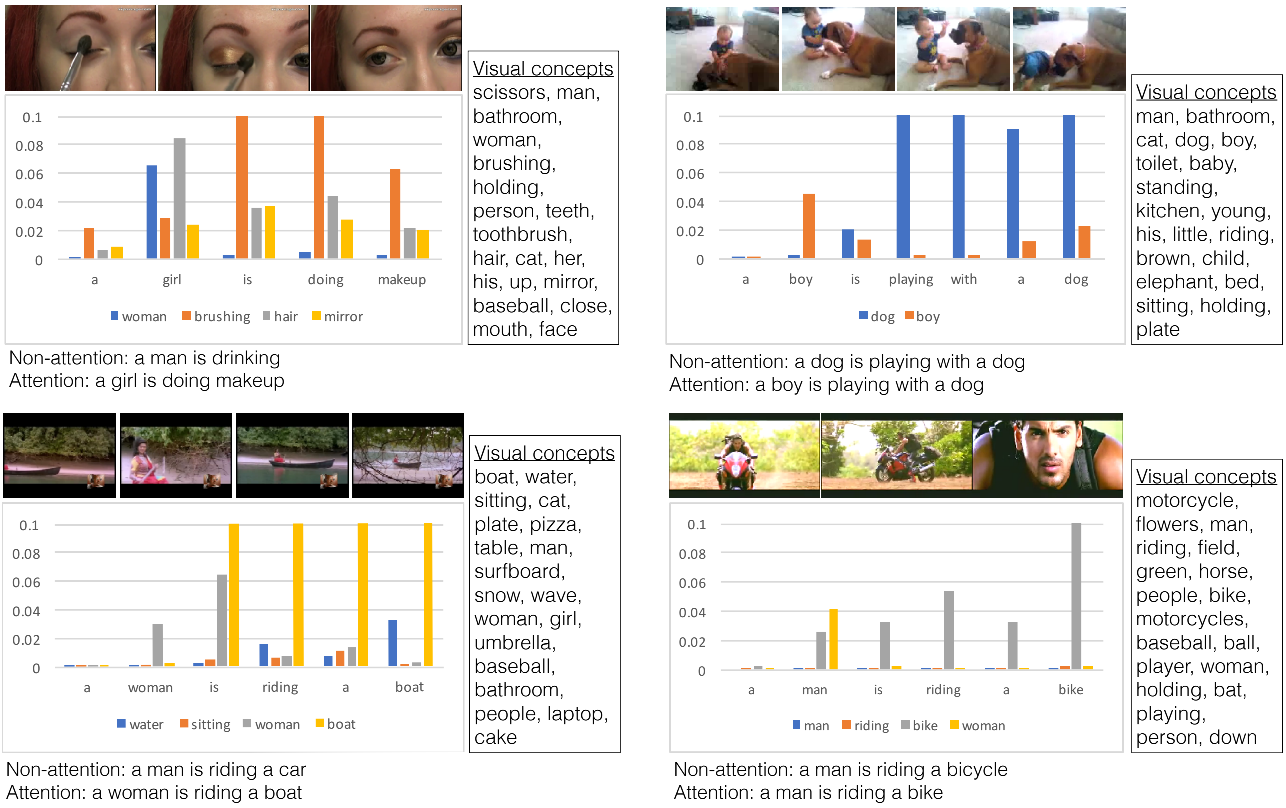
These metrics are common for evaluating image captioning and video description generation systems.

4.3. Experimental results

Table 1 shows the experimental results on MSVD dataset (Chen and Dolan, 2011). We compared our model to the sequence-to-sequence models reported by Venugopalan et al. (2015a) and Laokulrat et al. (2016), when using the same image features (VGG16). We can see some promising results in Figure 5, even though the semantic attention mechanism cannot clearly improve the scores of the test set. The relevant visual concepts were focused and the alignment weights changed properly when each word of the sentences were being generated. By focusing on visual concepts, the model can generate more precise mentions of the objects appearing in the scenes.

In the top-left example, the semantic attention model can recover the mis-mentioned word *(man)* to the correct word *(girl)* with high attention scores on the visual concepts *'woman'* and *'hair'*. It is also interesting that the model focuses on the concept *'brushing'* when correctly producing the phrase *'is doing make up'*. In the bottom-left example, with the help from the visual concept detector, the attention model can fix the error words *'man'* and *'car'* to the correct words *'woman'* and *'boat'*. In the top-right example, the semantic attention model can correctly mention the *'boy'*, while the non-attention model cannot. Lastly, in the bottom-right example, the mis-mentioned object *(bicycle)* in non-attention model can be correctly identified *(bike)* by the model with semantic attention. The visual concept *'bike'* was given a high attention weight when producing the word.

As we can see in Figure 5, many irrelevant visual concepts were detected. This is because the visual concept detector



Visual concepts
scissors, man, bathroom, woman, brushing, holding, person, teeth, toothbrush, hair, cat, her, his, up, mirror, baseball, close, mouth, face

Non-attention: a man is drinking
Attention: a girl is doing makeup

Visual concepts
man, bathroom, cat, dog, boy, toilet, baby, standing, kitchen, young, his, little, riding, brown, child, elephant, bed, sitting, holding, plate

Non-attention: a dog is playing with a dog
Attention: a boy is playing with a dog

Visual concepts
boat, water, sitting, cat, plate, pizza, table, man, surfboard, snow, wave, woman, girl, umbrella, baseball, bathroom, people, laptop, cake

Non-attention: a man is riding a car
Attention: a woman is riding a boat

Visual concepts
motorcycle, flowers, man, riding, field, green, horse, people, bike, motorcycles, baseball, ball, player, woman, holding, bat, playing, person, down

Non-attention: a man is riding a bicycle
Attention: a man is riding a bike

Figure 5: 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.

Model	BLEU	METEOR	CIDEr	ROUGE-L
System A				
Mean pooling	-	0.277	-	-
Seq-to-seq	-	0.292	-	-
Seq-to-seq + flow	-	**0.298**	-	-
System B				
Temp. attention	0.407	**0.310**	**0.615**	**0.676**
Ours				
Non-att.	0.430	**0.318**	**0.670**	0.616
Semantic-att.	**0.431**	0.317	0.668	**0.621**

Table 2: Scores of video description generation results on the MSVD dataset. System A is the results of the sequence-to-sequence model reported by Venugopalan et al (2015a). System B is the results of the sequence-to-sequence model with temporal attention by Laokulrat et al (2016). Note that, for System A, only METEOR scores were reported in the original paper.

was trained on another image dataset, so it could not perform well in video frames. We can apply this model to any kind of external information other than visual concepts.

5. Discussion

We have performed a deep analysis to find why the semantic attention model gets low scores even though it gives us very promising results. Below are the interesting points that we have found from the analysis.

Too specific description. The attention model tends to produce more specific descriptions and therefore is likely to be given lower scores. As shown in Figure 6 (a) and (b), in both examples, the sentences produced by the NON-ATT model[1] get perfect scores (BLEU=1.0, METEOR=1.0, ROUGE=1.0), while the sentences produced by the SEMANTIC-ATT model get much lower scores since they are not perfectly correct. The first video (Figure 6 (a)) shows *'a man pouring oil into a pan'*, and the second video (Figure 6 (b)) shows *'a man sprinkling spices into a pan'*.

Training data. The dataset contains bad examples. The sentence *'a man cooking his kichen'*[1] appears many times in the training data and the ground-truth sentences of the test data. So, if the model produces exactly this sentence, it will get a perfect score. See Figure 6 (a) and (b) for reference.

Furthermore, the misspelling of the training captions and the words outside pre-trained GloVe embeddings will both be converted to $\langle UNK \rangle$ tokens, which can worsen the learning of the models. We can fix this issue by correcting the misspelled words and re-training the word embeddings to cover our vocabularies.

Evaluation metrics. The evaluation metrics are not perfect. In Figure 6 (c), the sentence by the NON-ATT model gets a higher BLEU score even though it is wrong. Also, the

[1] Note that the grammatical and spelling errors are originally from the training data. For fair comparison with other previous work, we did not modify the data.

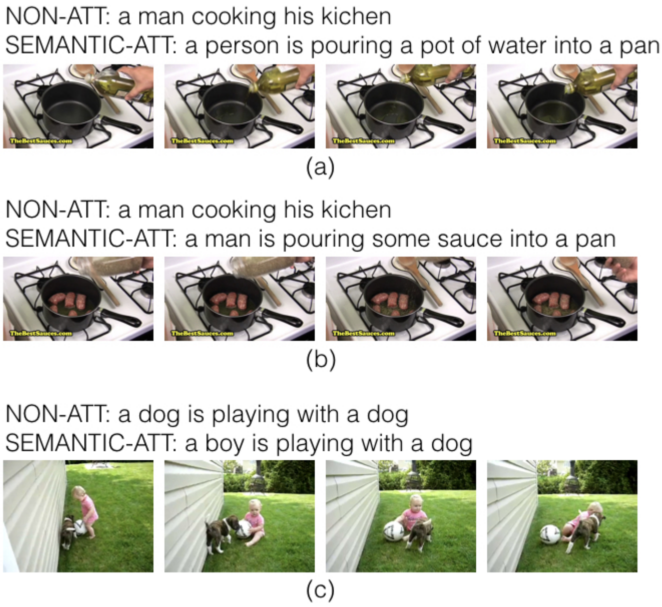

NON-ATT: a man cooking his kichen
SEMANTIC-ATT: a person is pouring a pot of water into a pan
(a)
NON-ATT: a man cooking his kichen
SEMANTIC-ATT: a man is pouring some sauce into a pan
(b)
NON-ATT: a dog is playing with a dog
SEMANTIC-ATT: a boy is playing with a dog
(c)

Figure 6: Videos used in the discussion. Read Section 5. for more detail[1].

word 'boy' does not appear in the ground-truth sentences. All of the ground-truth sentences use the word *'baby'*, so the score of the SEMANTIC-ATT model is even lowered.

For these reasons, the quantitative improvement is small and not obvious, but we believe that the results from the attention model are promising and potentially useful, especially when the visual concept detector can work well. We can replace the concept detector with other object/action prediction models, or combine collections of words detected by two or more detectors. Our semantic attention model can flexibly incorporate that external information into the conventional sequence-to-sequence model.

6. Conclusion

We have proposed a sequence-to-sequence model with semantic attention for video description generation, which can flexibly incorporate that external information into the conventional sequence-to-sequence model. The results show that the model is able to learn to focus on external fine-grained information of videos and a have better quality of video descriptions. The results from the attention model are promising and potentially useful, especially when the visual concept detector can work well. We can replace the concept detector with other object/action prediction models, or combine collections of words detected by two or more detectors.

Acknowledgements

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). We also would like to thank the anonymous reviewers for their insightful comments and suggestions, which were helpful in improving the quality of the paper.

7. Bibliographical References

Chen, D. L. and Dolan, W. B. (2011). Collecting highly parallel data for paraphrase evaluation. In *ACL 2011*.

Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. (2015). Microsoft COCO captions: Data collection and evaluation server. *arXiv:1504.00325*.

Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Fang, H., Gupta, S., Iandola, F. N., Srivastava, R. K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J. C., Zitnick, C. L., and Zweig, G. (2015). From captions to visual concepts and back. In *CVPR 2015*.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*.

Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *CVPR 2015*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv:1412.6980v8*.

Laokulrat, N., Phan, S., Nishida, N., Shu, R., Ehara, Y., Okazaki, N., Miyao, Y., and Nakayama, H. (2016). Generating video description using sequence-to-sequence model with temporal attention. In *COLING 2016*.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *ACL 2002*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP 2014*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Tokui, S., Oono, K., Hido, S., and Clayton, J. (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)*.

Vedantam, R., Zitnick, C. L., and Parikh, D. (2014). Cider: Consensus-based image description evaluation. *arXiv:1411.5726v2*.

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., and Saenko, K. (2015a). Sequence to sequence – video to text. In *ICCV 2015*.

Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., and Saenko, K. (2015b). Translating videos to natural language using deep recurrent neural networks. In *NAACL-HLT 2015*.

Venugopalan, S., Hendricks, L. A., Rohrbach, M., Mooney, R. J., Darrell, T., and Saenko, K. (2017). Captioning images with diverse objects. In *CVPR 2017*.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2014). Show and tell: A neural image caption generator. *arXiv:1411.4555*.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *arXiv:1502.03044v3*.

Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., and Courville, A. (2015). Describing videos by exploiting temporal structure. In *ICCV 2015*.

You, Q., Jin, H., Wang, Z., Fang, C., and Luo, J. (2016). Image captioning with semantic attention. In *CVPR 2016*.