Generating Video Description using Sequence-to-sequence Model with Temporal Attention

Natsuda Laokulrat[1], Sang Phan[2], Noriki Nishida[3], Raphael Shu[3], Yo Ehara[1],
Naoaki Okazaki[4,1], Yusuke Miyao[2,1] and Hideki Nakayama[3,1]

[1]Artificial Intelligence Research Center (AIRC), National Institute
of Advanced Industrial Science and Technology (AIST), Japan
[2]National Institute of Informatics (NII), Japan
[3]The University of Tokyo, Japan
[4]Tohoku University, Japan

Abstract

Automatic video description generation has recently been getting attention after rapid advancement in image caption generation. Automatically generating description for a video is more challenging than for an image due to its temporal dynamics of frames. Most of the work relied on Recurrent Neural Network (RNN) and recently attentional mechanisms have also been applied to make the model learn to focus on some frames of the video while generating each word in a describing sentence. In this paper, we focus on a sequence-to-sequence approach with temporal attention mechanism. We analyze and compare the results from different attention model configuration. By applying the temporal attention mechanism to the system, we can achieve a METEOR score of 0.310 on Microsoft Video Description dataset, which outperformed the state-of-the-art system so far.

1 Introduction

Since the recent breakthrough in machine learning, generating description for static images has been intensively researched and high-quality image description can be achieved in the past few years by many research groups (Vinyals et al., 2014; Karpathy and Fei-Fei, 2015; Fang et al., 2015; Xu et al., 2015; You et al., 2016). However, video description generation is a much more challenging task, which requires understanding temporal relationship between video frames. Automatically generating video description can be useful in many aspects. It can help visually-impaired people to understand the content of videos. More importantly, it will enable computers to understand videos, rather than just working at pixel levels, because the generated descriptions contain objects appearing the videos along with their attributes, locations, actions, and relations with other objects. Though considered very challenging, being able to understand videos can have great impact and will be useful to many other applications, such as human-robot interaction, video indexing and query, and video classification.

This paper focuses on generating video description using a encoder-decoder sequence-to-sequence model with temporal attention mechanism. We perform a set of experiments using different configurations of attention mechanisms and also can achieve state-of-the-art results. Our main contributions are as follows: First, we apply temporal attention mechanism to the encoder-decoder sequence-to-sequence model and are able to outperform the state-of-the-art system on Microsoft Video Description dataset (MSVD) (Chen and Dolan, 2011) in terms of the METEOR (Denkowski and Lavie, 2014), CIDEr (Vedantam et al., 2014), and ROUGE-L (Lin, 2004) scores. Second, we analyze and compare the results from different model configurations, and show how temporal attention works in generating sentences. We also performed the experiments on a large movie dataset, Montreal Movie Annotation Dataset (M-VAD) collected by Torabi et al. (2015).

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers,
pages 44–52, Osaka, Japan, December 11-17 2016.

2 Related Work

There exist many works in the image captioning task inspired by recent advances in machine translation using encoder-decoder Recurrent Neural Network (RNN) (Bahdanau et al., 2014). Vinyals et al. (2014) replaced the RNN encoder with a Convolutional Neural Network (CNN) which was pre-trained for an image classification task. Then, the last hidden layer of the pre-trained CNN can be used to produce a meaningful representation of an image, which they used as an input to the RNN decoder that generates sentences.

Karpathy and Fei-Fei (2015) used a combination of a CNN and a bidirectional RNN to generate natural language sentences from an image as well as their corresponding regions. Fang et al. (2015) first trained visual detectors from a dataset of image-caption pairs, and then used the output words from the visual detectors as input to the language model for generating image description. Xu et al. (2015) also used a CNN-RNN encoder-decoder scheme and applied a spatial attention mechanism over an input image, so that the model can attend to a specific region of an image while generating each word of a caption sentence. The model of You et al. (2016) learned to selectively attend to semantic concepts (similar to visual concepts in (Fang et al., 2015)) of an image and input them into the RNN decoder at each time step of generating image description.

After great success in image captioning, researchers are currently moving forward into working on generating sentences that describe videos. 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 description 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. Yao et al. (2015) has incorporated an attentional mechanism to video caption generation. They took into account both local and global temporal structures of videos by incorporating a spatial temporal 3D CNN.

A sequence-to-sequence model for generating description of videos has been first proposed by 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.

3 Sequence-to-sequence Model

This section describes the concept and structure of the sequence-to-sequence model, including Long Short-Term Memory (LSTM), the two-layer encoder-decoder LSTM, and the temporal attention mechanism that we used in this work.

3.1 Long Short-Term Memory

An LSTM network, proposed by Hochreiter and Schmidhuber (1997), is a type of RNN that is commonly used in sequence-to-sequence models. It has been intensively used in machine translation, speech recognition as well as in image/video description generation.

LSTM can help to avoid exploding and vanishing gradient problems by using forget gates to reset memory block when they are out of date. Given an input x_t, at time step t, one unit of an LSTM can be formulated as

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

where i_t, f_t and o_t are input gates, forget gates, and output gates. The symbol $*$ 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

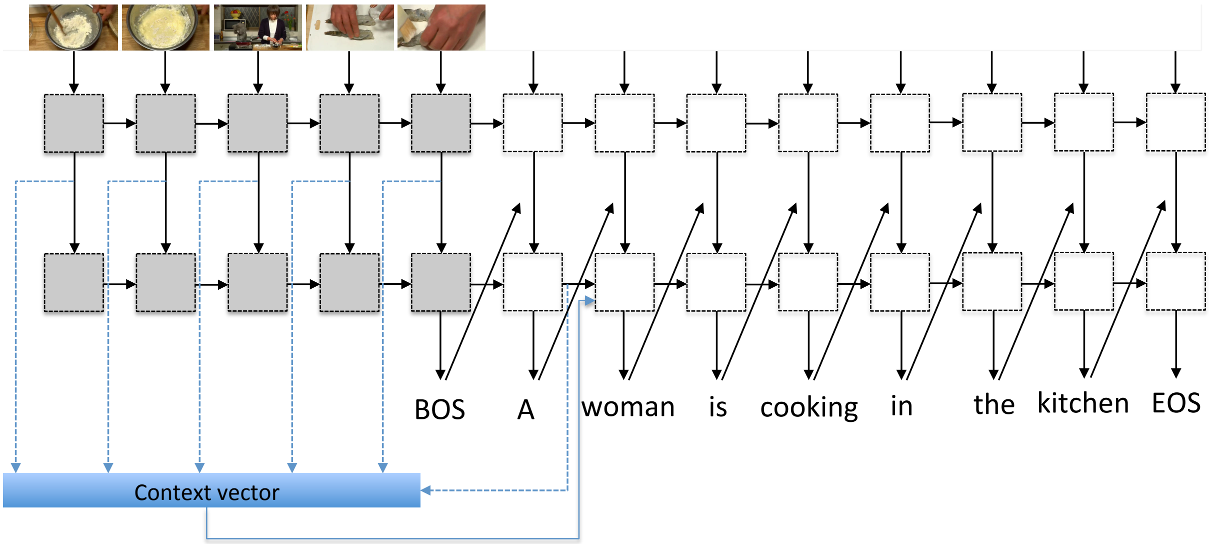


Figure 1: System architecture of the sequence-to-sequence model with temporal attention. In the figure, we omit the image embedding layer, the word embedding layer, and the softmax layer, due to the space constraint.

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.

3.2 Two-layer LSTM for Sequence-to-sequence Model

Figure 1 depicts our two-layer LSTM model for generating sentences from a video, which is based on the sequence-to-sequence model 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_i is the i^{th} frame of the video, we can formulate our system as

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

where $h_t^{(1)}$ is the hidden state of the first (upper) LSTM layer, defined as $LSTM^{(1)}$, at time step t. In the encoding stage, the input $x_t = v_t$ and, in the decoding stage, $x_t = \vec{0}$.

The input to the second (lower) LSTM layer is the concatenation of the word (represented as word embedding) generated on the previous time step $t-1$ and the hidden state of the first LSTM layer.

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

where $h_t^{(2)}$ is the hidden state of the second LSTM layer, defined as $LSTM^{(2)}$, at time step t. In the encoding stage, we fix the word $w_{t-1} = \vec{0}$, since there is no word being generated. Lastly, the distribution over all the words at time step t can be computed by

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

3.3 Temporal Attention Mechanism

Our approach incorporates the previously-proposed sequence-to-sequence model with a temporal attention mechanism. The second-layer LSTM at decoding stage can be formulated as

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

where the context vector c_t, at the time step t in the decoding stage, is the weighted sum of encoder's hidden states $h_i^{(1)}$.

$$c_t = \sum_{i=1}^{n} \alpha_i^{(t)} h_i^{(1)} \tag{6}$$

	videos	sentences	tokens	vocab size	avg. length	captions/video	source
MSVD	1,970	80,827	567,874	12,594	10.2s	≈40	crowd
M-VAD	46,589	55,904	502,926	17,609	6.2s	1	professional

Table 1: Video description dataset statistics. Refer to Venugopalan et al. (2015a) and Torabi et al. (2015) for more details.

The weight $\alpha_i^{(t)}$ is computed at every time step t and can be computed by

$$\alpha_i^{(t)} = \frac{e^{a(h_i^{(1)}, h_{t-1}^{(2)})}}{\sum_{j=1}^{n} e^{a(h_j^{(1)}, h_{t-1}^{(2)})}} \tag{7}$$

where $a(h_i^{(1)}, h_{t-1}^{(2)})$ is the alignment function used to calculate relevance scores between every hidden state $h_i^{(1)}$ in the encoding stage the hidden state $h_{t-1}^{(2)}$ at the previous time step $t-1$.

3.4 Alignment model

To see which alignment functions are suitable for the model, we have used four different alignment functions in this work. Three of them were used in Luong et al. (2015), and we also use summation of hidden states as an alignment function.

$$a(h_i^{(1)}, h_{t-1}^{(2)}) = \begin{cases} {h_i^{(1)}}^\top h_{t-1}^{(2)} & \text{dot} \\ {h_i^{(1)}}^\top W_a h_{t-1}^{(2)} & \text{bilinear} \\ W_a[h_i^{(1)}; h_{t-1}^{(2)}] & \text{concat} \\ W_a h_i^{(1)} + W_b h_{t-1}^{(2)} & \text{sum} \end{cases} \tag{8}$$

The parameters W_a and W_b of the alignment model are jointly learned at training time with all other parameters in the network.

4 Dataset and Experiment Setting

This section describes the video description datasets, the pre-processing steps, the experiment setting, and the evaluation metrics that we used.

4.1 Dataset and pre-processing

In this paper, we trained the models and generated descriptions for two publicly-available video datasets as follows. The summary of the datasets is shown in Table 1.

Microsoft Research Video Description Corpus (MSVD) collected by Chen and Dolan (2011). It is a set of video clips aggregated from Youtube, containing 1,970 short clips with ≈40 captions/per clip. The videos were collected and annotated by crowdsourcing on Amazon Mechanical Turk. The clips mostly contain a single activity and can be described using only one sentence. For fair comparison with other 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 1200, 100, and 670, respectively. We also use the pre-processed sentences and vocabularies from Venugopalan et al. (2015b). The pre-processing includes tokenizing, converting to lower case, and removing punctuations.

Montreal Video Annotation Dataset (M-VAD) M-VAD is a large collection of movie clips provided by Torabi et al. (2015). It was collected from 92 movies, and spitted into 46,589 short clips. Each clip is associated with a description, which can be more than one sentence. The dataset provides an official training/validation/test split, consisting of 36,921, 4,717 and 4,951 video clips respectively. We used all words in the training data as our vocabulary set and only pre-processed the data by tokenizing the sentences.

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**	-	-
Results reported by Yao et al. (2015)				
Enc-Dec (Basic)	0.387	0.287	0.448	-
+ Local (3-D CNN)	0.388	0.283	0.509	-
+ Global (temporal attention)	0.403	0.290	0.480	-
+ Local + Global	**0.419**	**0.296**	**0.517**	-
Our system (VGG16) - *non-attention*	0.381	0.300	0.562	0.654
Our system (VGG16) - *dot*	**0.411**	0.307	0.574	0.664
Our system (VGG16) - *bilinear*	0.407	**0.310**	**0.615**	**0.676**
Our system (VGG16) - *concat*	0.390	**0.310**	0.595	0.667
Our system (VGG16) - *sum*	0.385	0.306	0.584	0.664
Our system (ResNet) - *non-attention*	0.427	0.318	0.706	0.675
Our system (ResNet) - *dot*	0.406	***0.326**	***0.750**	0.680
Our system (ResNet) - *bilinear*	0.425	0.318	0.733	0.675
Our system (ResNet) - *concat*	0.417	0.325	0.723	***0.681**
Our system (ResNet) - *sum*	***0.437**	0.319	0.718	0.676

Table 2: Scores of video description generation results on the MSVD dataset. * marks the top scores of each column.

4.2 Experiment Setting

We down-sample all the video clips by selecting every 8^{th} frame from the original videos and resize them to 224x224. We extract features for each frame using the pre-trained image classification models provided publicly in *Caffe Model Zoo* (Jia et al., 2014). In this work, we performed the experiments using the features extracted from the 4096-dimensional fc7 layer of the 16-layer VGG model (VGG16), proposed by Simonyan and Zisserman (2014), and the 2048-dimensional output from Deep Residual Networks (ResNet), recently proposed by He et al. (2015), who is the winner in the ILSVRC 2015 classification task. We embed input frame features into 512-dimensional embeddings.

For text input, after pre-processing, the word tokens are represented by one-hot vectors. We use the word BOS to mark the beginning of a sentence and EOS to represent the end of the sentence. We also embed the word vectors into 512-dimensional embeddings. The parameters for both image and word embedding layers are jointly learned with other parameters at training time.

We fix the number of encoding and decoding time steps in order to enable batch training. For the MSVD dataset, we constrain the number of encoding and decoding time steps to be 60 and 20, respectively. The M-VAD corpus has longer sentence length, so we set the number time steps to 50 and 30 for encoding and decoding, respectively, to allow the language model to generate longer sentences.

In every experiment, the LSTM hidden layer size is set to 1,000. We use the Adam optimizer (Kingma and Ba, 2014) with the learning rate of 0.0001 and the mini-batch size of 40. We also apply the dropout strategy (Srivastava et al., 2014) with the ratio of 0.3 at the video input layer to avoid overfitting. We implemented our system using *Chainer*, which is a powerful framework for developing neural networks developed by Tokui et al. (2015).

5 Experimental Results and Discussion

This section shows the experimental results in both qualitative and quantitative aspects. We performed a quantitative analysis of results based on four evaluation metrics, including

Model	BLEU	METEOR	CIDEr	ROUGE-L
Results reported by Venugopalan et al. (2015a)				
Mean pooling (VGG16)	-	0.061	-	-
Sequence to sequence (VGG16)	-	**0.067**	-	-
Results reported by Yao et al. (2015)				
Enc-Dec (Basic)	0.003	0.044	0.044	-
+ Local (3-D CNN)	0.004	0.051	0.050	-
+ Global (temporal attention)	0.003	0.040	0.047	-
+ Local + Global	**0.007**	**0.057**	**0.061**	-
Our system (VGG16) - *non-attention*	***0.008**	***0.072**	0.087	***0.159**
Our system (VGG16) - *dot*	***0.008**	0.062	***0.088**	0.140
Our system (VGG16) - *concat*	0.006	0.067	0.082	0.143
Our system (VGG16) - *sum*	0.007	0.070	0.074	0.155

Table 3: Scores of video description generation results on the M-VAD dataset. * marks the top scores of each column.

- **BLEU** (Papineni et al., 2002), an evaluation metric widely used in machine translation. BLEU calculates a score based on modified n-gram precision of the generated sentence against a set of human-annotated reference sentences.

- **METEOR** (Denkowski and Lavie, 2014), an automatic metric for machine translation evaluation. It is based on explicit word-to-word matching between the generated sentence and one or more reference sentences. METEOR supports matching between words with simple morphological variants and synonyms.

- **CIDEr** (Vedantam et al., 2014), an automatic consensus metric of image description quality. Consensus-based Image Description Evaluation (CIDEr) measures the similarity of a computer-generated sentence against a set of human-annotated sentences. It gives a higher score to the sentence that is more similar to the majority of human written descriptions.

- **ROUGE-L** (Lin, 2004), a recall-oriented evaluation metric popularly used in summarization community. It measures the number of in-sequence unigram matches between the generated sentence and sentences created by annotators.

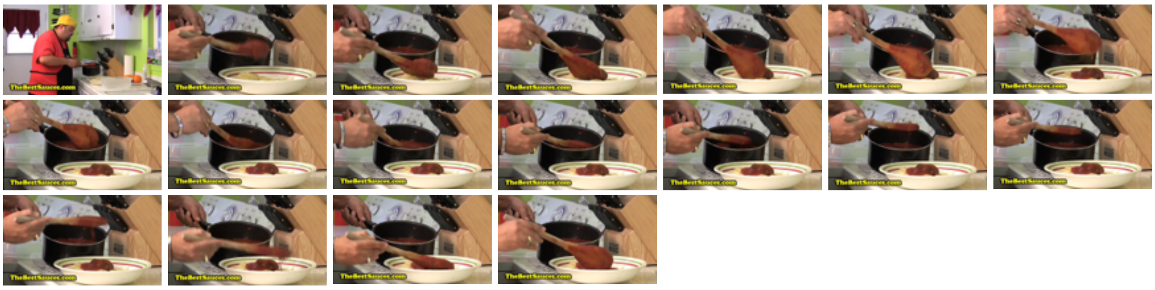
We use the caption evaluation package provided by the Microsoft COCO Image Captioning Challenge (Chen et al., 2015).

We compare our results to the results reported by the mean-pooling and sequence-to-sequence approaches reported in (Venugopalan et al., 2015a), and results from the CNN-RNN encoder-decoder model with a temporal attention mechanism reported in (Yao et al., 2015). However, the comparison to (Yao et al., 2015) is probably not a fair comparison, since we used different CNNs for feature extraction, e.g. they used GoogleNet and 3D-CNN but we used VGG16 and ResNet.

5.1 Results on the MSVD dataset

The results on the MSVD dataset are presented in Table 2 and the sample sentences are shown in Figure 2. The attention model plays an important role for both VGG16 and ResNet experiments. We can achieve the BLEU scores of 0.411 with VGG16 features and 0.437 with ResNet features. Our METEOR scores reached 0.310 when using VGG16 and 0.326 when using ResNet. However, in the experiment using VGG16 features, the *bilinear* alignment function seems to work best, while the *dot* alignment function gives the highest performance for the ResNet feature set.

As we can clearly see from the table, ResNet features are very powerful and can achieve the highest scores in all evaluation metrics.



VGG16
(non-attention) a man is adding sauce to a bowl
(dot) a man is adding sauce to a bowl
(bilinear) a man is pouring sauce into a bowl of chili
(concat) a man is pouring some sauce into a bowl
(sum) a man is pouring some sauce into a bowl

(ground truth)
(1) the man is pouring sauce over the pasta
(2) a man is putting food from pan to a plate
(3) a man is adding sauce to his spaghetti

ResNet
(non-attention) a man is pouring sauce into a pot
(dot) a man is stirring a pot of food
(bilinear) a man is pouring sauce into a pot
(concat) a person is adding water to a pot
(sum) a man is stirring a pot

Figure 2: Generated descriptions from MSVD dataset.

5.2 Results on the M-VAD dataset

The results on the M-VAD dataset are presented in Table 3. For this dataset, we did not perform the experiments using all the model configurations and feature types, due to the time constraints.

Even though the previous experiment on MSVD dataset showed that the result with ResNet were better than those with VGG16, we decided to use image features extracted from VGG16 to make a fair comparison with the previous work. From the table, our non-attention model can outperform the previous work in all evaluation metrics, but the attention model does not work well in this dataset. The reason probably comes from the characteristic of the M-VAD dataset that the videos contain a very high diversity of scenes and descriptions, so our attention models cannot be learned properly.

Some samples of generated sentences are shown in Figure 3.

6 Conclusion

In this paper, we have proposed a framework to automatically generate descriptions for video clips. We have applied the temporal attention mechanism to the sequence-to-sequence LSTM model. The results have proved that our model can generate high-quality short descriptions for videos, and can outperform the previous work. With the temporal attention mechanism, the model can learn to selectively focus on different parts of a video while generating each describing word.

For future work, we would like to use audio features or include a text-to-speech system in our framework since we think that audio is a very important piece of information for video understanding.

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.

VGG16

(non-attention) SOMEONE grabs a gun and the man steps out of the room .
(dot) SOMEONE and SOMEONE watch the men in their hands and the others .
(concat) He turns and walks off . SOMEONE follows SOMEONE to the floor , his eyes closed .
(sum) SOMEONE and SOMEONE step out of the room . SOMEONE and SOMEONE walk through the crowd .

(ground truth) SOMEONE appears and shoots SOMEONE in the leg . The mobster slips away . SOMEONE grabs SOMEONE .

Figure 3: Generated descriptions from M-VAD dataset.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*.

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*.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K. Srivastava, Li Deng, Piotr Dollar, 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*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *arXiv:1512.03385*.

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

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*.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR 2015*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980v8*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP 2015*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *ACL 2002*.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

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)*.

Atousa Torabi, Christopher J. Pal, Hugo Larochelle, and Aaron C. Courville. 2015. Using descriptive video services to create a large data source for video annotation research. *arXiv:1503.01070*.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2014. Cider: Consensus-based image description evaluation. *arXiv:1411.5726v2*.

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*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *arXiv:1502.03044v3*.

Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Describing videos by exploiting temporal structure. In *ICCV 2015*.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. *arXiv:1603.03925*.