Memory Augmented Recurrent Neural Networks for Dense Video Captioning

BMVC 2019 Submission # 594

Abstract

Dense video captioning is a challenging computer vision task that involves effectively understanding long video sequences. In this work, we address this problem by augmenting recurrent neural network architectures with external memory. We propose a dense captioning model that incorporates external memory augmentation both to encode video and densely caption it. We demonstrate that recurrent video encoder and dense captioner networks augmented with external memory can be used to effectively encode frames based on the content of the entire video, as well as for generating dense captions better than recurrent networks without external memory augmentation. We conduct experiments on the ActivityNet Captions and YouCook II datasets to demonstrate the potential of external memory augmentation.

1 Introduction

Describing the content of a video in natural language is a fundamental artificial intelligence problem with many applications, such as video search, video summarization, and acessibility for the visually impaired. In the task of dense video captioning, we are given video input that consists of multiple events, often chronologically related to each other, and the goal is to detect these events and describe each of them using a natural language sentence.

Numerous methods involving recurrent neural networks (RNNs) have been proposed to address this task [7, 10, 12]. RNNs can be used either for video encoding or captioning the events in these videos, or for both of these purposes. While RNNs are shown to be effective at sequence understanding, understanding long sequences is still a difficult problem.

We present a novel architecture for dense video captioning based on memory augmented neural networks. A large and sparsely written external memory can offer a potential benefit to recurrent nets in understanding long sequences [1, 3, 8]. Dense video captioning involves two main problems: dense event detection and dense captioning. In this work, we focus on dense captioning. Memory augmented recurrent neural networks have ideal properties for understanding long videos and densely captioning them. In particular, they enable the storage and access of memory cells that can capture the content of events as they evolve over varying timescales. Furthermore, cells in the external memory are written sparsely, which lets them store data reliably for long timescales, unlike the neurons in RNN models where whole neurons are updated at every iteration. This property provides memory augmented networks a mechanism to store information about the long sequence of events which would enable a contextual understanding.

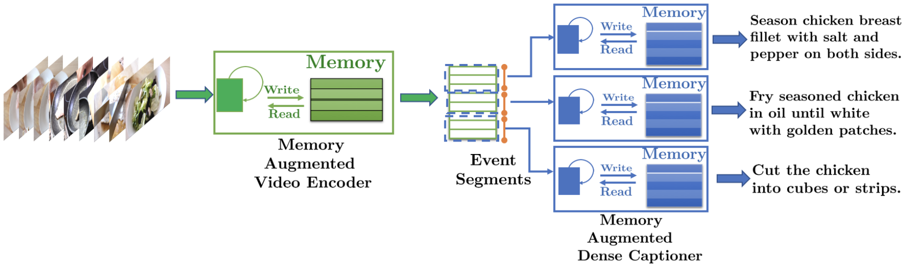

Figure 1: An overview of our memory augmented recurrent neural network based model for dense video captioning. We use a memory augmented recurrent representation to encode the whole video and also to caption each event segment.

As shown in Fig. 1, we use memory augmented recurrent neural networks for our model components. First, a memory augmented video encoder is used to produce a feature representation of event segments in a holistic manner based on context from other events occurring in the video. Second, each of the event segments is captioned coherently by a memory augmented network using these holistically learnt event segment representations. We demonstrate the effectiveness of this over baseline methods that do not have memory augmentation.

2 Related Work

Dense Video Captioning. Krishna et al. [6] proposed the ActivityNet Captions dataset that aims to benchmark event detection algorithms which can also provide a natural language description about these detected events. Zhou et al. [16] introduce a new procedural video dataset called YouCookII which contains YouTube cooking videos annotated densely with event segments and a natural language description about each of these events. Wang et al. [7] use a bi-directional recurrent representation to encode video frames for event detection and also to extract event context vectors for dense captioning. Xu et al. [12] learn to perform joint detection of events and describe them using 3D convolutional representation to detect events and a hierarchical LSTM representation to densely caption the video about these events. Zhou et al. [17] address dense captioning by using transformer based end-to-end event segment detection and captioning model with multiple layers of multi-headed self attention. Li et al. [7] propose to jointly learn to detect and caption the events by using "descriptiveness" regression to refine the segment boundaries and caption using an attribute augmented captioning architecture. Wang et al. [10] propose an adaptive bidirectional context fusion based on a gating mechanism for both the event proposal and event caption generation. Zhang et al. [15] utilize cross-modal hierarchical sequential embedding that learns multi-granular correspondances between image/video and text for performing different tasks including dense video captioning. Unlike many of the previous approaches for this task that rely on recurrent neural network representations to function both as a memory bank and video-caption representation learners, in our model, we provide dedicated stable external memories both for our video generator and captioner.

Image and Video Captioning is a computer vision task that has an extensive body of literature behind it. Some of the earlier examples of image and video captioning methods include Donahue et al. [2] that uses an encoder decoder technique to caption images and videos. You et al. [13] invoke semantic attribute attention maps on both the input and output caption representations to learn a captioning model. Karpathy et al. [5] propose to describe images by inferring a latent image to caption alignment by performing multimodal embedding and using a structured objective. Yu et al. [14] learn a hierarchical representation for paragraph captioning of video by incorporating temporal and spatial attention mechanisms.

Memory models are frequently used to learn sequence representations for performing various tasks like question answering in text [3], movie question answering [8] and image captioning [1]. Graves et al. [3] introduce a novel external memory augmented recurrent neural network to perform multiple tasks that include question answering in synthetic dataset settings, and three graph processing based tasks. Na et al. [8] use a write CNN to encode multimodal data content and a read CNN to read this content as well as the question to learn the answer representation. Park et al. [1] learn a personalized image captioning representation by involving a memory which is used as a storage bank to capture contextual data representations that are pertinent to hashtag prediction and post generation which are primary tasks with in this work. Wang et al. [11] propose a multimodal memory for video captioning. Differing from these methods, we focus on the task of dense video captioning which involves generating a caption for each of the events in videos.

3 Method

The proposed model consists of a recurrent external memory aided video encoder and caption generator. We first provide a brief description about external memory augmented neural networks and follow it up with a description about our dense captioning model aided by this network.

3.1 Preliminaries

Inspired by Graves et al. [3], our memory encoder consists of an external memory that enhances the storage capacity of recurrent neural networks and a memory controller that accesses this memory to store and retrieve history. We provide an overview of each of these components and their functionalities.

3.1.1 Memory Controller

The memory controller consists of a recurrent neural network that uses the external memory to store information. Iteratively, it reads and writes to the external memory, and in this process, it encodes the temporal dynamics of the input. The controller encoded input representation comprises of the controller recurrent neural network output, and in addition, a set of “read vectors" being read from the external memory. Mathematically, the controller operation can be described as:

$$o_t = C(f_t; M^r_{t-1}); M^r_t \tag{1}$$

Here, C denotes a controller recurrent network, f_t, M^r_t are the feature input and external memory read vectors at time t. “;" is the concatenation operator. The read vectors are obtained from the controller's read to the external memory. The controller performs iterative read/write operations on the external memory, described later in this section and in the supplementary material. The controller network emits this output o_t. The final encoded representation is obtained by computing a residual connection over o_t at time t.

$$\tilde{o}_t = o_t + g(f_t) \tag{2}$$

Here, function g maps input f_t to the output space. $\tilde{o}_t$ is the final output of the network.

3.1.2 External Memory

The external memory consists of a block of memory cells used by the controller to store memory. At each timestep, the controller reads a set of data from the memory and writes another set of data to the memory. The memory read/write locations are governed by a probabilistic addressing mechanism, including content based addressing. In addition to the content based addressing, additional addressing components specific to read/write operations are also utilized to serve customized functionalities accordingly. Next, we provide a summary of these addressing mechanisms. For more mathematical details of its operation, please refer to the supplementary.

At each timestep t, content based addressing chooses the most appropriate location to perform the operation by computing a probability map over locations. It uses a key vector k and computes cosine similarity between this key vector and content at memory locations:

$$c_t = softmax(cos(M_{t-1}, k_t)\tilde{s}_t) \tag{3}$$

Here, c_t denotes content weight for memory locations in M_t, and $\tilde{s}_t$ denotes “strength" value constrained to a range between $[1, \infty)$, all at time t. At each timestep, content based addressing uses a read and write key/strength pair to perform read/write operation. We now briefly describe the read/write operation. For a more detailed mathematical description about the read write operations, please refer to the supplementary.

Write Operation: The write operation involves computing the content (known as “write vector") and the location to write to the memory (known as “write weightings"), determined by a set of differentiable components. Intuitively, the write location is determined by parameters that choose between re-writing a location that has been written by **content based addressing** (Eqn. 3) and writing to a new location by a technique called **dynamic memory addressing**. In addition to this component, the write operation also enables an operation of preventing any write operation using a learnable “write" gate and to trigger a memory reset using an “erase" vector. The mathematical formulation of the write process involving these components can be found in the supplementary material.

Read Operation: Similar to the write operation, the read operation involves computing the location to read, known as “read weightings". A read operation is determined by multiple factors. Following Equation 3, the content based addressing weight component corresponding to read weights is computed for memory locations. In addition to the content based addressing component, a read operation also consists of a component that tracks the temporal order in which the contents are recorded in the memory. To do so, first, a **precedence vector** is computed which measures frequency of write operations at a given location. Second, using this precedence vector, a temporal **linkage matrix** is computed, which is used in encoding the write location order (known as temporal links) in the form of **forward** and **backward** link weights. Final read weightings are computed as a weighted combination of these three components. Mathematical formulation of the read process involving these components can be found in the supplementary material.

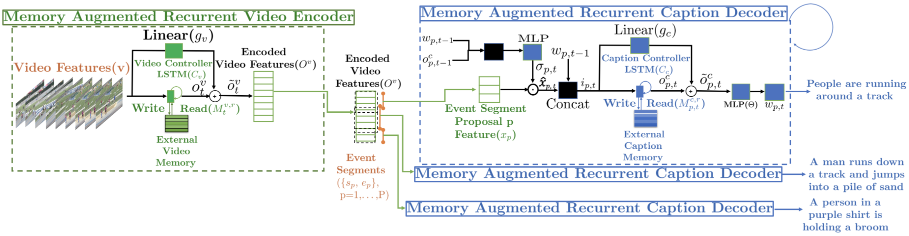

Figure 2: Illustration of our dense captioning model with references to variables in our equations. Encoder components are shown in green and decoder components are shown in blue.

3.2 Memory Augmented Recurrent Video Encoder

In this work, we are given an input video that contains multiple segments of interest that need to be captioned, for which we employ a memory augmented representation to encode the video in a holistic manner and to caption all the events that occur in the video. A detailed illustration of our model is shown in Fig. 2. We now describe each of these components.

We use a memory augmented recurrent neural network to encode the multi-event video in a holistic manner. The recurrent neural network in our video controller, C_v of our video encoder is a single layer bidirectional LSTM (Bi-LSTM). Each frame in the input v_t is encoded using this representation as:

$$o_t^v = (C_v(v_t; M_{t-1}^{v,r}); M_t^{v,r}) \tag{4}$$

where $M_t^{v,r}$ are a set of read vectors from the timestep t, and o_t^v is the output of video controller network. We further compute a transformation over residual connection [4] on controller to get the final encoded video representation:

$$\tilde{o}_t^v = o_t^v + g_v(v_t) \tag{5}$$

Here, g_v is a function that maps video input to the video encoder network's output space. We refer to the entire encoded video representation as O^v, which has features corresponding to T video frames stacked together:

$$O^v = (\tilde{o}_1^v, \tilde{o}_2^v, \dots, \tilde{o}_T^v) \tag{6}$$

3.3 Memory Augmented Recurrent Dense Caption Generator

We use the encoded video representation O^v to caption the events that occur in the video. The input to this module is the encoded video representation O_v and a set of P event segments. An event segment p is defined as a tuple (s_p, e_p) of its start and end locations. Using these inputs, we first extract event segment features as:

$$x_p = (O_{s_p}^v, O_{s_p+1}^v, \dots, O_{e_p}^v) \tag{7}$$

We caption these segments x_p using our dense captioning model.

We enable the event captioner to focus on different parts of the event segment appropriately while generating the caption word-by-word, using temporal attention. At each word generation step, event segment features are temporally attended to compute the video feature input to the caption decoder at time t, denoted by $\hat{x}_{p,t}$:

$$\hat{x}_{p,t} = \sigma_{p,t} \cdot x_p \tag{8}$$

Here, $\sigma_{p,t}$ is the temporal attention weight vector for the p^{th} event segment at time t, and "$\cdot$" denotes inner product. We use this attended event segment representation to the caption network, which we use to compute the attention weights $\sigma_{p,t}$ described later.

We use a representation consisting of a memory augmented recurrent neural network for the caption controller network. It consists of a single layer Bi-LSTM, which is used to decode the caption word by word. We use this network in decoding the next word of the caption as follows:

$$i_{p,t} = [w_{p,t-1}; \hat{x}_{p,t}] \tag{9}$$

$$o^c_{p,t} = [(C_c(i_{p,t}; M^{c,r}_{p,t-1})); M^{c,r}_{p,t}] \tag{10}$$

$$\tilde{o}^c_{p,t} = o^c_{p,t} + g_c(i_{p,t}) \tag{11}$$

$$w_{p,t} = softmax(\theta(\tilde{o}^c_{p,t})) \tag{12}$$

In the above equations, p and t correspond to event segment and time indices respectively, $i_{p,t}$ is the input to the caption controller network. C_c is the Bi-LSTM neural network part of the caption controller, and $M^{c,r}_{p,t}$ are a set of read vectors at time t. The function g_c maps input $i_{p,t}$ to the caption controller network's output space and is used to compute the output word representation. We use the output of the caption controller network $\tilde{o}^c_{p,t}$ to decode the next word $w_{p,t}$ by computing a non-linear transformation θ followed by a softmax operation. Note from Equation 11 that similar to the video encoder, the captioning network too computes a residual connection across the caption controller. Using caption controller network outputs from the last timestep, the video attention weight $\sigma_{p,t}$ used in Equation 8 is recursively computed using the controller hidden state as:

$$\sigma_{p,t} = softmax(\phi([o^c_{p,t-1}; w_{p,t-1}])) \tag{13}$$

Here, $o^c_{p,t}$ is the caption controller network output defined in Equation 10, and $w_{p,t-1}$ is the previous caption word generated using Equation 12. ϕ is a linear transformation map that is used to compute the final video attention weight $\sigma_{p,t}$.
The output of dense caption generator is then a set of captions corresponding to each event segment:

$$((w^1_1, w^1_2, \ldots, w^1_{L^1_c}), (w^2_1, w^2_2, \ldots, w^2_{L^2_c})), \ldots, (w^p_1, w^p_2, \ldots, w^p_{L^p_c}), \ldots, (w^P_1, w^P_2, \ldots, w^P_{L^P_c})) \tag{14}$$

Here, index p corresponds to event segment index, and $L^1_c, L^2_c, \ldots, L^P_c$ represent the length of P captions, corresponding to each event segment respectively.

In summary, we propose a novel dense video captioning model consisting of a memory augmented video encoder and memory augmented dense caption generator. We generate captions, one corresponding to each event in the video, independently of each other, by performing temporal attention over encoded event segment features.

4 Training

To learn our model, we are provided with a training set with ground truth event segments and a caption associated with each event segment. Mathematically, each instance in the training set is represented by:

$$t_n = \{(s^i, e^i, g^i), i = 1, 2, \dots, P_n\} \tag{15}$$

Here, a training datapoint t_n, indexed by n, has an annotated set of P_n event segments, and each event segment annotation consists of start time s^i, end time e^i and a caption g^i.

We train the model end to end, and use cross entropy loss over words across all the P_n captions as our loss function for this training example:

$$Loss = \sum_{i=1}^{P_n} \sum_{t=1}^{L_c^i} CE(w_t^i, g_t^i) \tag{16}$$

Here, g_t^i denotes the t^{th} ground truth word of the i^{th} caption, and L_c^i denotes the length of the i^{th} caption. For a video, we compute this loss as a summation over all captions. We perform teacher forcing over the entire duration of training, where we input the ground truth caption word at each time t instead of the word predicted by the model. During test time, we input the previously predicted word instead of the ground truth.

5 Experiments

5.1 Datasets

We conduct experiments on two datasets: the YouCookII [16] and the ActivityNet Captions [6] datasets. The YouCook II dataset has 2000 videos, with 1333 videos for training and 457 videos for validation. The videos in this dataset have an average event count of 7.70. The ActivityNet Captions dataset has 10k training videos and 4917 validation videos. The videos have an average event count of 3.65 in this dataset. We use ResNet-34 features provided in case of YouCookII dataset, and C3D features in case of ActivityNet Captions dataset. We perform experiments using validation set as test data.

5.2 Model Settings

We use two external memory augmented recurrent neural networks, one for video encoding, one for captioning, with the same parameter dimensions for both the networks. We use an external memory of 5 memory cells with size 1024. We have 4 read heads that result in 4 read vectors at each timestep, and 1 write head. For the controller, we use a Bi-LSTM for the video, and an LSTM cell for the caption controller, each with size 1024. We vary the learning rate between 0.1 and 0.01 with step size of 2 upon attaining training error plateau. We train our system for a fixed training time of 50 epochs. We report maximum BLEU and METEOR scores that we obtain for each of our baseline methods and the model.

We report results obtained using the ground truth event segments. We restrict ourselves to ground truth event segments when comparing with previous methods as our models focus on the dense captioning task and do not perform end-to-end training with an event detector as in some previous work [10, 17].

5.3 Baselines

LSTM Captioner/video encoder: We use Bi-LSTM for the video encoder and LSTM for the captioner, both without the external memory. We refer to this baseline as LSTM-vid/LSTM-cap in the results section. This is the baseline method for our model variants.

LSTM Captioner, Memory augmented LSTM video encoder: We use Bi-LSTM with the external memory for the video encoder and LSTM with no external memory for the captioner. We refer to this model variant as Mem-Vid/LSTM-cap.
Memory augmented Captioner, Memory augmented LSTM video encoder: We use Bi-LSTM with the external memory for both video encoder and captioner. We refer to this model variant as Mem-Vid/Mem-cap.

5.4 Results

Tab. 1 lists the performance of several methods including our model on the YouCook II dataset using ground truth events. We obtain state of the art performance on the BLEU 4 metric and also achieve better performance compared to the baseline methods. Tab. 2 compares our method with previous dense video captioning methods applied to this problem and other dense video captioning/video captioning methods. We show that our model achieves competitive performance compared to these previous methods.

Method	BLEU 4	METEOR
LSTM-vid/LSTM-cap	0.93	9.15
Ours-Mem-Vid/LSTM-cap	1.49	9.74
Ours-Mem-Vid/Mem-cap	**1.64**	10.08
Zhou et al. [17][1]	1.42	**11.2**

Table 1: Experimental results on YouCookII dataset obtained using ground truth event segments.

Method	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR
LSTM-YT [9]	18.40	8.76	3.99	1.53	8.66
HRNN [14]	18.41	8.80	4.08	1.59	8.81
Krishna et al. [6]	18.13	8.43	4.09	1.60	8.88
Li et al. [7]	19.57	9.90	4.55	1.62	10.33
Zhang et al. [15]	19.8	9.4	4.3	2.1	9.2
LSTM-vid/LSTM-cap	18.91	7.75	3.09	1.55	8.95
Ours-Mem-Vid/LSTM-cap	**21.75**	**10.06**	4.30	1.92	9.76
Ours-Mem-Vid/Mem-cap	21.67	9.87	4.15	1.90	9.84
Zhou et al. [17][1]	-	-	**5.80**	**2.77**	**11.2**

Table 2: Experimental results on ActivityNet Captions dataset for all our methods using ground truth event segments (Numbers for the first four rows obtained from Li et al. [7]).

Relative to previous methods, our model achieves much more competitive performance in the case of the YouCookII dataset. The plausible reason for this observation lies in the nature of these two datsets. While YouCookII has events that are sequentially highly correlated events, ActivityNet videos have more independent events. The former scenario is more favourable to our model, as it aims to capture these long term correlations. Nevertheless, our models achieve competitive performance on both the datasets.

[1] Uses different features

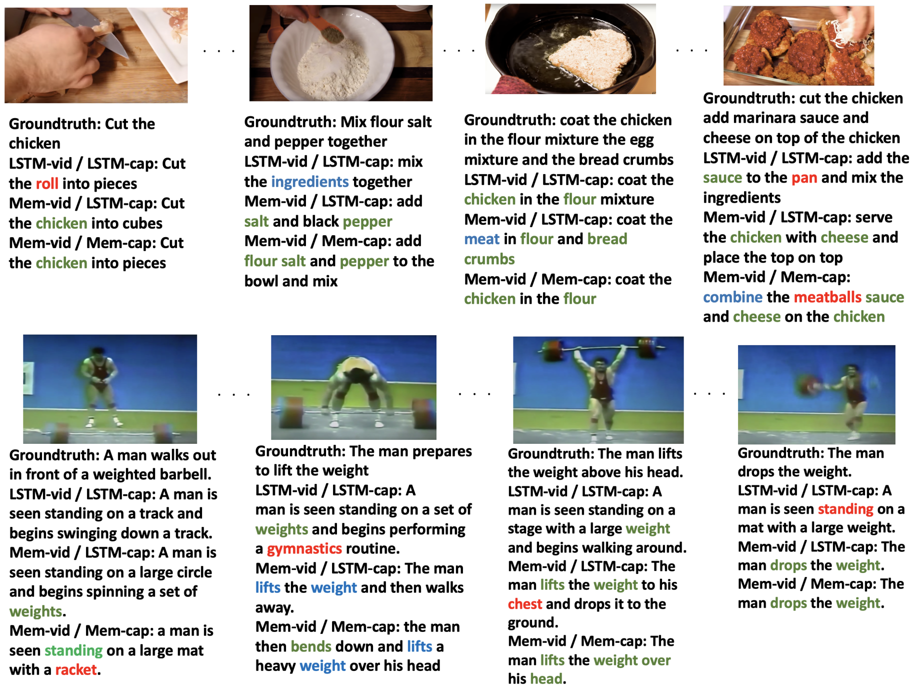

Figure 3: Sample qualitative results comparing ground truth captions with baseline and model variants. Note that the full model is able to generate the relevant content, shown in green (for exact attributes) and blue (for related attributes), while making fewer mistakes, shown in red.

Fig. 3 shows qualitative results and compares the results of different baselines. It can be seen that the memory augmented models refer to the relevant attributes of the event more often than the LSTM only baseline. This shows that memory augmentation improves the performance of recurrent models for dense captioning, and therefore could be extended to previous dense captioning methods involving recurrent neural network representations [7, 10].

6 Conclusion

In this work, we proposed a new model for dense video captioning involving an external memory augmented video encoder and an external memory augmented dense captioner. We showed that our model considerably improves the performance of recurrent neural network based dense captioning method, and is competitive with respect to previous state of the art dense captioning methods on two datasets.

References

[1] Cesc Chunseong Park, Byeongchang Kim, and Gunhee Kim. Attend to you: Personalized image captioning with context sequence memory networks. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, July 2017.

[2] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[3] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471, 2016.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[5] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[6] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE Conference on Computer Vision*, pages 706–715, 2017.

[7] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Jointly localizing and describing events for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7492–7500, 2018.

[8] Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. A read-write memory network for movie story understanding. In *Proceedings of the International Conference on Computer Vision*, 2017.

[9] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[10] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[11] Junbo Wang, Wei Wang, Yan Huang, Liang Wang, and Tieniu Tan. M3: multimodal memory modelling for video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7512–7520, 2018.

[12] Huijuan Xu, Boyang Li, Vasili Ramanishka, Leonid Sigal, and Kate Saenko. Joint event detection and description in continuous video streams. *arXiv preprint arXiv:1802.10250*, 2018.

[13] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4651–4659, 2016.

[14] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4584–4593, 2016.

[15] Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *Proceedings of the European Conference on Computer Vision*, pages 374–390, 2018.

[16] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. *arXiv preprint arXiv: 1703.09788*, 2017.

[17] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018.