Memory-Attended Recurrent Network for Video Captioning

Wenjie Pei[1], Jiyuan Zhang[1], Xiangrong Wang[2], Lei Ke[1], Xiaoyong Shen[1] and Yu-Wing Tai[1]*
[1]Tencent, [2]Southern University of Science and Technology
wenjiecoder@outlook.com, mikejyzhang@tencent.com, x.wang-2@tudelft.nl
keleiwhu@gmail.com, goodshenxy@gmail.com, yuwingtai@tencent.com

Abstract

Typical techniques for video captioning follow the encoder-decoder framework, which can only focus on one source video being processed. A potential disadvantage of such design is that it cannot capture the multiple visual context information of a word appearing in more than one relevant videos in training data. To tackle this limitation, we propose the Memory-Attended Recurrent Network (MARN) for video captioning, in which a memory structure is designed to explore the full-spectrum correspondence between a word and its various similar visual contexts across videos in training data. Thus, our model is able to achieve a more comprehensive understanding for each word and yield higher captioning quality. Furthermore, the built memory structure enables our method to model the compatibility between adjacent words explicitly instead of asking the model to learn implicitly, as most existing models do. Extensive validation on two real-word datasets demonstrates that our MARN consistently outperforms state-of-the-art methods.

1. Introduction

Video captioning aims to generate a sequence of words to describe the visual content of a video in a style of natural language. It has extensive applications such as Visual Question Answering (VQA) [28, 64], video retrieval [63] and assisting visually-impaired people [49]. Video captioning is a more challenging problem than its twin 'image captioning', which has been widely studied [1, 35, 48, 60]. This is not only because video contains substantially more information than still image, but it is also crucial to capture the temporal dynamics to understand the video content as a whole.

Most existing methods to video captioning follow the encoder-decoder framework [12, 19, 23, 26, 31, 34, 39, 50, 61], which employs an encoder (typically performed by CNNs or RNNs) to analyze and extract useful visual context features from the source video, and a decoder to generate the caption sequentially. The incorporation of attention mechanism into the decoding process has dramatically improved the performance of video captioning due to its capability of selective focus on the relevant visual content [12, 23, 52, 61]. One potential limitation of the encoder-decoder framework is that the decoder can only focus on one source video which is currently being processed while decoding. This implies that it can only investigate the correspondence between a word and visual features from a single video input. However, a candidate word in the vocabulary may appear in multiple video scenes that have similar but not identical context information. Consequently existing models cannot effectively explore the full spectrum between the word and its various similar visual contexts across videos in training data. For instance, the basis decoder in Figure 1, which is based on encoder-decoder framework, cannot corresponds the action in the source video to the word 'pouring' accurately because of insufficient understanding about the candidate word 'pouring'.

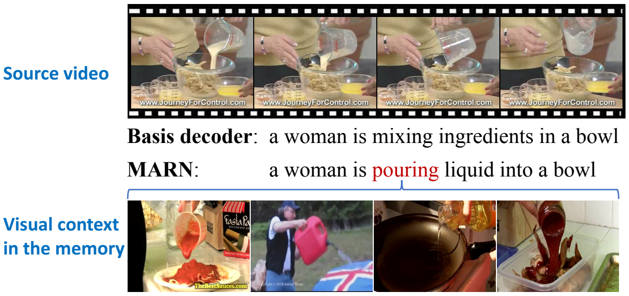
Source video
Basis decoder: a woman is mixing ingredients in a bowl
MARN: a woman is pouring liquid into a bowl
Visual context in the memory

Figure 1. The typical video captioning models based on the encoder-decoder framework (e.g., the Basis decoder in this figure) can only focus on one source video being processed. Thus, it is hard to explore the comprehensive context information about a candidate word, like 'pouring'. In contrast, our proposed MARN is able to capture the full-spectrum correspondence between the candidate word ('pouring' in this example) and its various similar visual contexts (all kinds of pouring actions) across videos in training data , which yields more accurate caption.

Inspired by the memory scheme which is leveraged to incorporate the document context in document-level machine translation [14], in this paper we propose a novel *Memory-Attended Recurrent Network* (MARN) for video captioning which explores the captions of videos with similar visual

[*] Corresponding author is Yu-Wing Tai

contents in training data to enhance the quality of generated video caption. Specifically, we first build an attention-based recurrent decoder as the basis decoder, which follows the encoder-decoder framework. Then we build a memory structure to store the descriptive information for each word in the vocabulary, which is expected to build a full spectrum of correspondence between a word and all of its relevant visual contexts appearing in the training data. Thus, our model is able to obtain a more comprehensive understanding for each word. The constructed memory is further leveraged to perform decoding using an attention mechanism. This memory-based decoder can be considered as an assistant decoder to enhance the captioning quality. Figure 1 shows that our model can successfully recognize the action 'pouring' in the source video because of the full-spectrum contexts (various pouring actions) in the memory.

Another benefit of MARN is that it can model the compatibility between two adjacent words explicitly. This comes in contrast to the conventional method adopted by most existing models (based on recurrent networks), which learns the compatibility implicitly by predicting the next word based on the current word and context information. We evaluate the performance of MARN on two popular datasets (MSR-VTT [59] and MSVD [5]) of video captioning. Our model achieves the best results comparing with other state-of-the-art video captioning methods.

2. Related Work

Video Captioning. Traditional video captioning methods are mainly based on template generation which utilizes the word roles (such as subject, verb and object) and language grammar rules to generate video caption. For instance, the Conditional Random Field (CRF) are employed to model different components of a source video [36] and then generate the corresponding caption in a way of machine translation. Also hierarchical structures are utilized to either model the semantic correspondences between concepts of actions and the visual features [22] or learn the underlying semantic relationships between different sentence components [13]. Nevertheless, these methods are limited in modeling the language semantics in captions due to the strong dependence on the predefined template.

As a result of rapid development of deep learning including convolutional networks (CNNs) and recurrent networks (RNNs), the encoder-decoder framework was first introduced by MP-LSTM [47], which employs CNNs as encoder to extract visual features from source videos and then decodes captions by LSTM. Another classical benchmark model based on encoder-decoder framework is S2VT [46], which shares a LSTM in both encoder and decoder. Subsequently, the attention mechanism gives rise to a significant performance boost to video captioning [61].

Recently, state-of-the-art methods based on encoder-decoder framework seek to make a breakthrough either in the encoding phase [6, 10, 30, 34, 56] or in the decoding phase [37, 51, 62]. Take for examples the cases that focus on the encoding phase, VideoLAB [34] proposes to fuse multiple modalities of source information to improve the captioning performance while PickNet [6] aims to pick the informative frames by reinforcement learning. TSA-ED [56] proposes to extract the spatial-temporal representation at trajectory level using attention mechanism. In the cases that focus on the decoding phases, RecNet [51] refines the captioning by reconstructing visual features from decoding hidden states and Aalto [37] designs a evaluator to pick the best caption from multiple candidate captions.

Most of these methods suffer from a potential drawback that the decoder can only focus on one source video being processed. Hence, they cannot capture the multiple visual context information of a candidate word appearing in rich video context in training data. Our proposed MARN, while following the encoder-decoder framework, is able to mitigate this limitation by incorporating the memory mechanism in the decoding phase to obtain a comprehensive understanding for each candidate word in the vocabulary.

Memory-based Models. Memory networks were first proposed to rectify the drawback of limited memory of recurrent networks (RNNs) [40, 55], which was then extended for various tasks. These memory-based models can be roughly grouped into two categories: serves as an assistant module [11, 14, 27, 57] or dominant module [8, 17, 29, 53]. In the first category, the memory is leveraged to assist the basis module to enhance the performance of the target task. For instance, two memory components are used to help the basis module, a sentence-based NMT, to capture the document context for document-level machine translation [14]. In the second category, the memory serves as a dominant module to perform the target task. An typical example is that memory networks is employed as the backbone for aspect sentiment classification [53]. Our proposed MARN falls into the first category since the memory is used as a assistant decoder in our video captioning system. To the best of our knowledge, our MARN is the first to leverage memory network in visual captioning.

3. Memory-Attended Recurrent Network

Our Memory-Attended Recurrent Network (MARN) consists of three modules: *encoder*, *attention-based recurrent decoder*, and *attended memory decoder*. The overall architecture of MARN is shown in Figure 2. After extracting effective features from the source video by the encoder, the attention-based recurrent decoder serves as a basis captioning decoder. Subsequently, the attended memory decoder is designed to enhance the captioning quality as an assistant decoder. We will first introduce the encoder and the attention-based recurrent decoder, then we will elaborate on the proposed attended memory decoder.

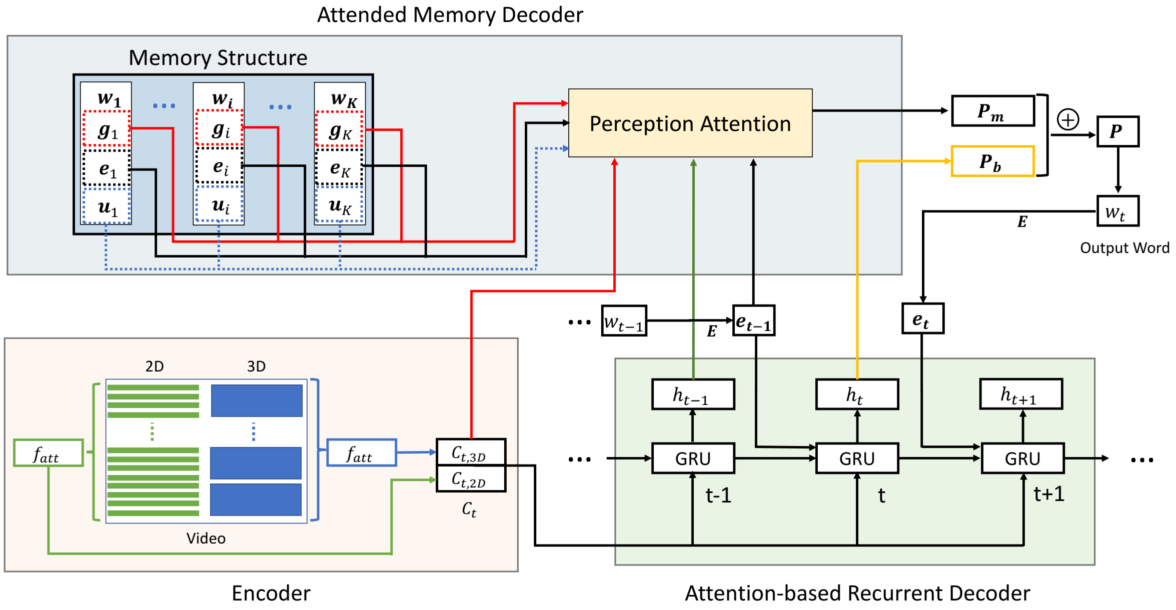

Figure 2. The architecture of our Memory-Attended Recurrent Network (MARN). It consists of three components: (1) Encoder for extracting features (both 2D and 3D) from the source video, (2) Attention-based Recurrent Decoder which is used as the basis captioning decoder and (3) Attended Memory Decoder which serves as an assistant decoder to enhance the captioning quality.

3.1. Encoder

The role of the encoder is to extract visual features from the input source video, which will be fed to the downstream decoder. A typical way is to employ pre-trained deep CNNs , such as GoogleNet [42, 61], VGG [38, 59] or Inception-V4 [41, 51], to extract 2D features for each of sampled images from the source video. Similarly, we also rely on the deep CNNs to extract 2D visual features. In our implementation, we opt for the ResNet-101 [16] pre-trained on imagenet [9] as the 2D-feature extractor of our encoder due to the its excellent performance and relatively high cost-efficiency. Furthermore, we also extract 3D visual features from the source video to capture the temporal information, which has been shown to be effective in vision tasks involving videos [23, 43]. Specifically, we employ the ResNeXt-101 [58] with 3D convolutions pretrained on Kinetics dataset [20] to extract 3D features, which has shown its superior performance on video classification tasks [15].

Formally, given a sequence of video frames $X = \{x_1, x_2, \ldots, x_L\}$ of length L, the 2D visual features obtained by pretrained ResNet-101 for each frame are denoted as $F_{2D} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_L\}$ in which $\mathbf{f}_l \in \mathbb{R}^d$. Besides, the 3D visual features are extracted by pretrained ResNeXt-101 for every 16 frames, i.e., the temporal resolution for each 3D feature is 16 frames. The resulting 3D features are denoted as $F_{3D} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$, where $N = L/16$ and $\mathbf{v}_n \in \mathbb{R}^c$. The obtained 2D and 3D visual features are then projected into hidden spaces with the same-dimension m by linear transformations:

$$\mathbf{f'}_l = \mathbf{M}_f\mathbf{f}_l + \mathbf{b}_f, \mathbf{v'}_n = \mathbf{M}_v\mathbf{v}_n + \mathbf{b}_v. \qquad (1)$$

Herein, $\mathbf{M}_f \in \mathbb{R}^{m \times d}$ and $\mathbf{M}_v \in \mathbb{R}^{m \times c}$ are transformation matrices while $\mathbf{b}_f \in \mathbb{R}^m$ and $\mathbf{b}_v \in \mathbb{R}^m$ are bias terms.

3.2. Attention-based Recurrent Decoder

The Attention-based Recurrent Decoder is designed as a basis decoder to generate the caption for the source video based on the visual features obtained from the Encoder. We adopt the similar model structure as Soft-Attention LSTM (SA-LSTM) [61]. A recurrent neural network is utilized as the backbone of the decoder to generate the caption word by word due to its powerful capability of modeling the temporal information by the recurrent structure. We use GRU [7] in our implementation (it is straightforward to replace it with LSTM in our MARN model). Meanwhile, the temporal attention mechanism is performed to make the decoder focus on the relevant (salient) visual features when generating each word by automatically learning attention weights for each frame of features.

Concretely, the t-th word prediction is performed as a classification task during the decoding process, which calculates the probability of a predicted word w_k among a vocabulary of size K via a softmax function:

$$P_b(w_k) = \frac{\exp\{\mathbf{W}_k\mathbf{h}_t + b_k\}}{\sum_{i=1}^{K} \exp\{\mathbf{W}_i\mathbf{h}_t + b_i\}}, \qquad (2)$$

where $\mathbf{W}_i$ and b_i refer to the parameters calculating the linear mapping score for i-th word in the vocabulary and $\mathbf{h}_t$ is the learned hidden state of GRU at the t-th time. Herein, $\mathbf{h}_t$ is achieved by GRU operations which take into account the hidden state in the previous step $\mathbf{h}_{t-1}$, the visual context information $\mathbf{c}_t$ and the word embedding of the predicted word in the previous step $\mathbf{e}_{t-1}$:

$$\mathbf{h}_t = \text{GRU}(\mathbf{h}_{t-1}, \mathbf{c}_t, \mathbf{e}_{t-1}), \tag{3}$$

where the embedding $\mathbf{e}_{t-1} \in \mathbb{R}^{d'}$ corresponds to the indexed vector in the embedding matrix $\mathbf{E} \in \mathbb{R}^{d' \times K}$. The temporal attention mechanism is applied to assign the attention weights for each frame of visual features, including both 2D and 3D features extracted by Encoder. Specifically, the context information of 2D visual features at t-th time step is calculated by:

$$\mathbf{c}_{t,2D} = \sum_{i=1}^{L} a_{i,t} \mathbf{f'}_i, \quad a_{i,t} = f_{att}(\mathbf{h}_{t-1}, \mathbf{f'}_i), \tag{4}$$

where L is the length of 2D visual features and f_{att} is the attention function which we adopt the same way as SA-LSTM [61]: a two-layer perceptron with $\tanh$ activation function in-between. We model the context information of 3D visual features $\mathbf{c}_{t,3D}$ in the similar way:

$$\mathbf{c}_{t,3D} = \sum_{i=1}^{N} a'_{i,t} \mathbf{v'}_i, \quad a'_{i,t} = f_{att}(\mathbf{h}_{t-1}, \mathbf{v'}_i), \tag{5}$$

and we obtain the final context information $\mathbf{c}_t$ by concatenating them together:

$$\mathbf{c}_t = [\mathbf{c}_{t,2D}; \mathbf{c}_{t,3D}]. \tag{6}$$

We share the attention function f_{att} in both 2D and 3D cases since it is able to guide the optimization of $\mathbf{M}_f$ and $\mathbf{M}_v$ in Equation 1 to project both 2D and 3D features into the similar feature space. It can be considered as a regularization to avoid potential overfitting compared to using two independent attention functions.

It should be noted that our model is different from previous methods utilizing both 2D and 3D visual features in the way how to aggregate them. Instead of simply fusing them together by concatenation in the early stage, we treat them separately during encoding and fuse their hidden representations by the attention mechanism in the decoding stage. A key benefit of this design is that the 2D and 3D features would not be inter-polluted, which is a typical problem as they represent different domains of visual features.

3.3. Attended Memory Decoder

We propose the Attended Memory Decoder as an assistant decoder to enhance the quality of the generated caption by the basis decoder (the Attention-based Recurrent Decoder). The rationale behind this design is that a word in the vocabulary may appear in multiple similar video scenes. While the attention-based decoder can only focus on one video scene while decoding, our attended memory decoder is designed to capture the full-spectrum context information from different video scenes where the same candidate word appears and thereby yielding a more comprehensive context for this word. Besides, the conventional attention-based decoder predicts the next word based on current word and context information instead of modeling the compatibility between two adjacent words explicitly. Our Attended Memory Decoder is expected to tackle this issue.

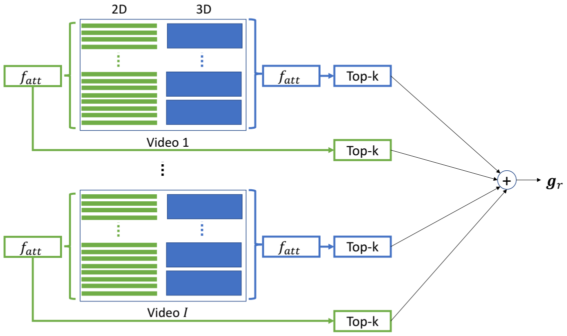

Figure 3. The visual context information $\mathbf{g}_r$ for word w_r is constructed by taking into account the top-k relevant frames from each of the related videos.

3.3.1 Memory Structure

The memory is designed to store the descriptive information for each word in the vocabulary. It is constructed as a mapping structure, among which each item is defined as a mapping from a word 'w' to its description 'd' : $\langle w, d \rangle$. In particular, the description 'd' consists of three components: 1) visual context information, 2) word embedding and 3) auxiliary features.

Visual context information. We extract the visual context information for a given word to describe its corresponding (salient) visual features contained in source videos by attention mechanism similar to Equation 4 and 5. Since a word may appear in multiple video scenes, we extract the salient visual features for each of the videos the word is involved in. To reduce the redundancy of extracted features, we only retain the top-k relevant features for each related video. As shown in Figure 3, the visual context information $\mathbf{g}_r$ for the r-th word in the vocabulary is modeled as:

$$\mathbf{g}_r = \frac{\sum_{i=1}^{I} \sum_{j=1}^{k} (a_{i,j} \mathbf{f'}_{i,j})}{\sum_{i=1}^{I} \sum_{j=1}^{k} a_{i,j}} + \frac{\sum_{i=1}^{I} \sum_{j=1}^{k} (a'_{i,j} \mathbf{v'}_{i,j})}{\sum_{i=1}^{I} \sum_{j=1}^{k} a'_{i,j}}, \tag{7}$$

where I is the number of related videos to the r-th word; $a_{i,j}$ and $a'_{i,j}$ are the j-th attention weights among the top k weights for 2D and 3D visual features respectively. Both 2D and 3D context features are normalized to make the magnitude of context features consistent for words with different frequencies. To avoid repetitive modeling, a straightforward way is to train the Attention-based Recurrent Decoder first and then reuse its attention module to extract visual context information for the memory.

Word embedding. The learned word embedding $\mathbf{e}_r$ of word w_r is also integrated into the memory module to quantitatively describe its properties such as semantics and syntactic features. It is readily achieved once the Attention-based Recurrent Decoder is trained.

Auxiliary features. The memory for a word is mainly constructed by the visual context information and word embedding. Additionally, we can also incorporate other potentially-useful auxiliary features, which is denoted as $\mathbf{u}_r$. For instance, we add the category information of videos (when it is available) in the memory, which can help to roughly cluster the video scenes and thereby assisting the decoding process.

Overall, the memory element corresponding to word w_r is represented as a map structure:

$$\langle w_r, d_r \rangle = \langle w_r, \{\mathbf{g}_r, \mathbf{e}_r, \mathbf{u}_r\}\rangle. \tag{8}$$

3.3.2 Decoding by Memory

The constructed memory is leveraged to build a caption decoding system, whose captioning results are further combined with the generated captions by the basis decoder (Attention-based Recurrent Decoder) to improve the captioning quality.

Specifically, we design the memory-attended decoding system as an attention mechanism upon the backbone of attention-based recurrent decoder. Similar to Equation 2, the probability that word w_k is predicted at the t-th time step is modeled via a softmax function:

$$P_m(w_k) = \frac{\exp\{q_k\}}{\sum_{i=1}^{K}\exp\{q_i\}}, \tag{9}$$

where K is the vocabulary size and q_i is the relevance score for word w_i which is used to measure the qualification of word w_i for the t-th time step based on its memory content. There are multiple ways to model the relevance score. We model it as a simple two-layer perceptron structure:

$$q_i = \mathbf{v}^\top \tanh\Big([\mathbf{W}_c \cdot \mathbf{c}_t + \mathbf{W}_g \cdot \mathbf{g}_i] + [\mathbf{W'}_e \cdot \mathbf{e}_{t-1} + \mathbf{W}_e \cdot \mathbf{e}_i] + \mathbf{W}_h \cdot \mathbf{h}_{t-1} + \mathbf{W}_u \cdot \mathbf{u}_i + \mathbf{b}\Big), \tag{10}$$

where $\mathbf{c}_t$, $\mathbf{e}_{t-1}$, $\mathbf{h}_{t-1}$ are respectively the context information at time step t, predicted word and hidden state at time step $t-1$ from the Attention-based Recurrent Decoder; $\mathbf{W}_c$, $\mathbf{W}_g$, $\mathbf{W'}_e$, $\mathbf{W}_e$, $\mathbf{W}_h$, $\mathbf{W}_u$ are the linear transformation matrices and $\mathbf{b}$ is the bias term.

The physical interpretation behind this modeling is: based on the current situation represented by $\mathbf{h}_{t-1}$, the term $[\mathbf{W}_c \cdot \mathbf{c}_t + \mathbf{W}_g \cdot \mathbf{g}_i]$ measures the compatibility between the visual context information of the current source video and the visual context information of the candidate word w_i; $[\mathbf{W'}_e \cdot \mathbf{e}_{t-1} + \mathbf{W}_e \cdot \mathbf{e}_i]$ measures the compatibility between the previously predicted word and the candidate word w_i; the term $\mathbf{W}_u \cdot \mathbf{u}_r$ corresponds to the auxiliary features.

Integrated caption decoding by MARN. With Attention-based Recurrent Decoder being the decoding basis and Attended Memory Decoder as the assist, our proposed Memory-Attended Recurrent Network (MARN) models the probability of the word w_k being the next one in the captions as:

$$P(w_k) = (1-\lambda)P_b(w_k) + \lambda P_m(w_k), \tag{11}$$

where λ is introduced to balance the contribution from two decoders. In practice, the value of λ is tuned on a held-out validation set.

3.4. Parameter Learning

Suppose we are given a training set $\mathcal{D} = \{x^{(n)}_{1,\dots,L^{(n)}}, w^{(n)}_{1,\dots,T^{(n)}}\}_{n=1,\dots,N}$ containing N videos and their associated captions. $L^{(n)}$ and $T^{(n)}$ are respectively the lengths of videos and captions for the n-th sample. Since the construction of the Memory relys on the Attention-based Recurrent Decoder, it is trained first and the Attended Memory Decoder is trained subsequently.

3.4.1 Attention-based Recurrent Decoder

Video captioning models are typically optimized by minimizing the negative log likelihood:

$$L_c = -\sum_{n=1}^{N}\sum_{t=1}^{T^{(n)}} \log P_b(w_t^{(n)}|x^{(n)}_{1,\dots,L}). \tag{12}$$

Attention-Coherent Loss (AC Loss) The visual attention weights learned in Equation 4, which is for constructing the context information, always fluctuates significantly even for the adjacent frames since they are learned independently. However, we believe that the attention weights should proceed smoothly. Besides, the attention weights, which are assigned to the frames in the time interval corresponding to a event or an action, should be close to each other. It is also consistent with the scheme of human attention. To this end, we propose a so-called Attention-Coherent Loss (AC Loss) to regularize the attention weights in Equation 4:

$$L_a = \sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{i=2}^{L} |a_{i,t}^{(n)} - a_{i-1,t}^{(n)}|, \tag{13}$$

which minimizes the gap between the attention weights for adjacent frames. Note that the AC Loss is not performed for 3D visual feature because each of the 3D visual features describes the a 3D voxel with a much higher temporal resolution (16 frames in our case) rather than a single frame. Therefore, the smoothness of the attention weights is not required.

Consequently, the Attention-based Recurrent Decoder is trained by minimizing the combined loss:

$$L = L_c + \beta L_a, \tag{14}$$

where β is a hype-parameter to balance two losses and is tuned on a held-out validation set.

3.4.2 Attended Memory Decoder

Similarly, the Attended Memory Decoder is optimized by minimizing the negative log likelihood:

$$L = -\sum_{n=1}^{N}\sum_{t=1}^{T^{(n)}} \log P_m(w_t^{(n)}|x^{(n)}_{1,\dots,L}). \tag{15}$$

4. Experiments

We conduct experiments to evaluate the performance of the proposed MARN on two benchmark datasets of video captioning: Microsoft Research-Video to Text (MSR-VTT) [59] and Microsoft Research Video Description Corpus (MSVD) [5]. We aim to (1) investigate the effect of Attended Memory Decoder on the performance of video captioning and (2) compare our MARN with the state-of-the-art methods for video captioning.

4.1. Datasets

MSR-VTT. MSR-VTT dataset is a widely-used benchmark dataset for video captioning. To have a fair comparison with previous methods, we use the initial version of MSR-VTT, which contains 10,000 video clips from 20 general categories. Each video clip is provided with 20 human-annotated natural sentences (captions) for reference collected by Amazon Mechanical Turk (AMT) workers. We follow the standard data split [59]: 6513 clips for training, 497 clips for test and the left 2990 clips for test.

MSVD. MSVD dataset contains 1970 short video clips collected from YouTube. Each video clip depicts a single activity and is annotated with 40 captions. Following the data split in previous work [61, 45, 50], 1200 video clips are held out for training, 100 clips for validation and 670 for test.

4.2. Experimental Setup

We construct the vocabulary based on the training set by filtering out words occurring fewer than three times, resulting in vocabularies of around 11K words and 4K words for MSR-VTT and MSVD, respectively.

The dimension of the word embedding is set to 512. For the GRU in the Attention-based Recurrent Decoder, the number of hidden units is set to 512. For the Encoder, we first extract 2D and 3D features with 2048 dimensions, and then transform them linearly into 512 dimensions as described in Equation 1. The dimensions of the attention modules in the Attention-based Recurrent Decoder and the Attended Memory Decoder are both tuned by selecting the best configuration from the option set $\{256, 384, 512\}$ using a held-out validation set.

We employ Adam [21] gradient descent optimization with gradient clipping between -5 and 5 [4]. We perform training for both decoders for 500 epochs with the learning rate decayed by 0.5 every 50 epochs. The final performance is determined by the trained model that performs best on the validation set. To compare our model with state-of-the-art methods, we adopt the standard automatic evaluation metrics, namely CIDEr [44], METEOR [3], ROUGE-L [25] and BLEU [33]. We use CIDEr, which is especially designed for captioning, as the evaluation metric in our ablation experiments, i.e., investigation on the effect of Attended Memory Decoder and Attention-Coherent Loss.

Model			Dataset	
Basis decoder	Memory	AC Loss	MSR-VTT	MSVD
√	×	×	45.7	89.9
√	√	×	46.8	91.7
√	√	√	**47.1**	**92.2**

Table 1. Performance measured by CIDEr (%) of our video captioning system equipped with different modules on both MSR-VTT and MSVD datasets (%) for ablation study. *Memory* refers to the Attended Memory Decoder.

Memory			Dataset	
Word embedding	Visual context	Auxiliary feature	MSR-VTT	MSVD
×	×	×	45.7	89.9
√	×	×	46.1	90.7
√	√	×	46.6	**91.7**
√	√	√	**46.8**	–

Table 2. Performance measured by CIDEr (%) of our video captioning system equipped with different components of the memory on both MSR-VTT and MSVD datasets for ablation study. *Auxiliary feature* refers to the category information in this experiment. Note that AC Loss is not used for all experiments here. The category information is not available for MSVD dataset.

4.3. Ablation Study

We first perform quantitative evaluation to investigate the effect of Attended Memory Decoder and Attention-Coherent Loss respectively. To this end, we conduct ablation experiments which begins with sole basis decoder, namely Attention-based Recurrent Decoder in the captioning system and then incrementally augments the system with Attended Memory Decoder and Attention-Coherent Loss. Table 1 presents the experimental results.

Effect of Attended Memory Decoder. Comparing the performance of sole basis decoder with integrated system of basis decoder and Attended Memory Decoder presented in Table 1, we observe that the Attended Memory Decoder boosts the performance of video captioning by 1.1% and 1.8% (in CIDEr) on MSR-VTT and MSVD, respectively. They are indeed substantial improvements considering the progresses reported in recent years by state-of-the-art methods on video captioning (refer to Table 3 and 5), which validates the effectiveness of our Attended Memory Decoder.

The memory is composed of three components: visual context, word embedding and auxiliary feature (As explained in Section 3.3.1). To further investigate the contribution from each of them to the whole system, we perform ablation study on the memory structure. The experimental results presented in Table 2 show that the word embedding and the visual context bring about the major performance boost while the auxiliary feature (category information) yields another minor gain on MSR-VTT dataset. The word embedding is used for measuring the compatibility between the previously predicted word and current candidate word while the visual context information is responsible for providing a full-spectrum context and measuring how close the candidate word matches the source video. Note that any extra information that is available and potentially helpful for captioning can be readily used as the auxiliary feature.

Effect of Attention-Coherent Loss. Table 1 shows the performance of the system with and without the proposed AC Loss. In particular, the performance is improved by a small margin for both datasets (from 46.8 to 47.1 for MSR-VTT and from 91.7 to 92.2 for MSVD).

4.4. Qualitative Evaluation of Attended Memory Decoder

To gain more insight into what MARN has learned in the memory and the effect of Attended Memory Decoder, we present several examples to qualitatively compare our MARN model with the basis decoder (Attention-based Recurrent Decoder) in Figure 4. Compared to the basis decoder, the MARN is able to decode more precise captions for the given source video, which benefits from the designed Attended Memory Decoder. Take Figure 4 (a) as an example, the basis decoder provides a reasonable caption for the video. However, it cannot recognize 'baby stroller' while our MARN successfully recognize it due to various 'baby stroller' in its memory corresponding to the word 'stroller'.

4.5. Comparison with Other Methods

Next, we compare our model with existing methods for video captioning on both MSR-VTT and MSVD datasets. All four popular evaluation metrics including CIDEr, METEOR, ROUGE-L and BLEU are reported. It should be noted that our model is not compared to the video captioning methods based on reinforcement learning (RL) [24, 54], which follows the routine setting in image captioning that the RL-based methods are evaluated separately from other methods (with no RL) [1, 18] for a fair comparison. Nevertheless, it is straightforward to extend our model using RL by applying Self-Critical Sequence Training [35] which is widely adopted in image captioning.

4.5.1 Comparison on MSR-VTT

We compare with two groups of baseline methods: 1) fundamental methods including S2VT [46] which shares a LSTM structure in both encoding and decoding phases, Mean-Pooling LSTM (MP-LSTM) [47] which performs a mean-pooling for all sampled visual frames as the input for a LSTM decoder and Soft-Attention LSTM (SA-LSTM) [61] which employs attention model to summarize visual features for decoding each word; 2) newly published state-of-the-art methods including RecNet [51] which refines the captioning by reconstructing the visual features from decoding hidden states, VideoLAB [34] which proposes to fuse source information of multiple modalities to improve the performance, PickNet [6] that picks the informative frames based on a reinforcement learning framework, Aalto [37] that designs a evaluator model to pick the best caption from multiple candidate captions, and ruc-uva [10] which proposes to incorporate tag embeddings in encoding while designing a specific model to re-rank the candidate captions.

Model	BLEU-4	METEROR	ROUGE-L	CIDEr
S2VT [46]	31.4	25.7	55.9	35.2
MP-LSTM (VGG19) [51]	34.8	24.7	–	–
SA-LSTM (VGG19) [51]	35.6	25.4	–	–
SA-LSTM (Inception-V4) [51]	36.3	25.5	58.3	39.9
RecNet$_{local}$ [51]	39.1	26.6	59.3	42.7
VideoLAB [34]	39.1	27.7	60.6	44.1
PickNet (V+L+C) [6]	**41.3**	27.7	59.8	44.1
Aalto [37]	39.8	26.9	59.8	45.7
ruc-uva [10]	38.7	26.9	58.7	45.9
Basis decoder (ours)	40.1	27.7	60.4	45.7
MARN (ours)	40.4	**28.1**	**60.7**	**47.1**

Table 3. Performance of different video captioning models on MSR-VTT dataset in terms of four metrics (%).

In Table 3 we show results on MSR-VTT dataset. Our proposed MARN achieves the best performance in terms of METEOR, ROUGE-L and CIDEr while ranking second on BLEU-4. This strongly indicates the superiority of our model. The fact that SA-LSTM outperforms S2VT or MP-LSTM validates the contribution of attention mechanism. Besides, the SA-LSTM equipped with Inception-V4 performs better than its variant with VGG19, which shows the importance of encoding scheme for visual features. The state-of-the-art models typically performs much better than the classical models such as SA-LSTM or MP-LSTM due to all kinds of techniques they proposed. Another interesting observation is that our basis model achieves comparable results with these state-of-the-art models, which somewhat implies the performance ceiling using only encoder-decoder framework and attention mechanism.

MARN vs Basis decoder		
Win	Tie	Loss
43.3%	23.3%	33.3%

Table 4. Human evaluation for comparing our MARN model with the basis decoder on a subset of MSR-VTT test set.

Human evaluation. As a complement to the standard evaluation metrics, we also performs a human evaluation to compare our model with the basis decoder. Specifically, we randomly select a subset from MSR-VTT test sets and ask 30 human subjects to make a comparison between the generated captions by our models and the basis decoder independently. We aggregate evaluation results of all subjects for each sample. Table 4 shows that our model wins among 43.3% test samples and fails on 33.3% samples against the basis decoder, which indicates the advantages of our model.

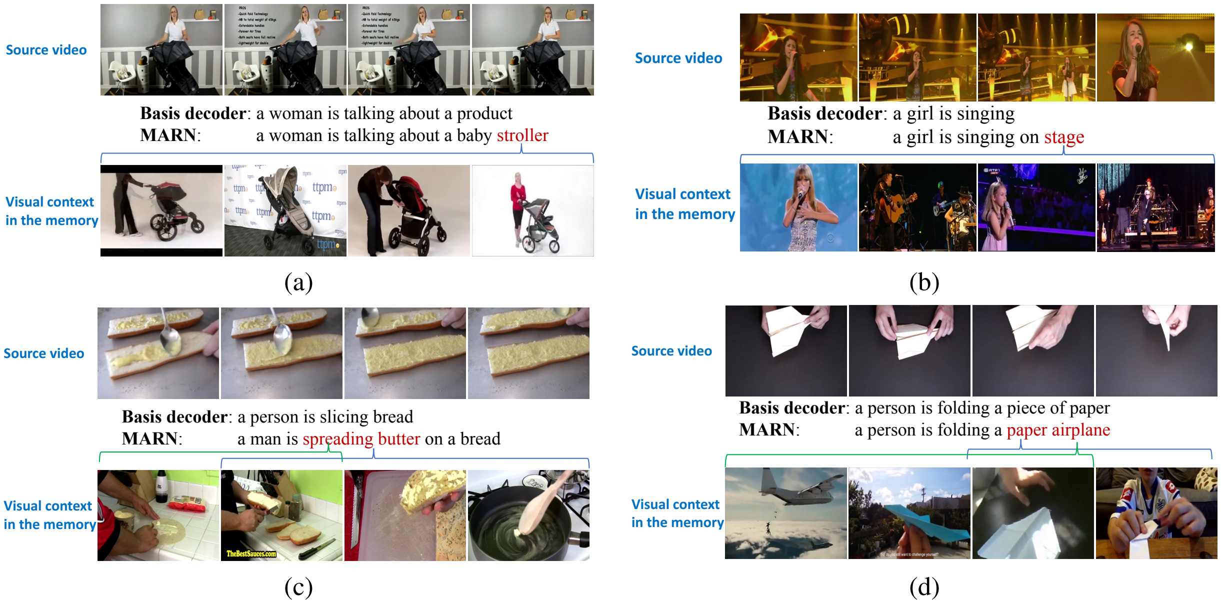
Source video
Basis decoder: a woman is talking about a product
MARN: a woman is talking about a baby stroller
Visual context in the memory
(a)
Source video
Basis decoder: a girl is singing
MARN: a girl is singing on stage
Visual context in the memory
(b)
Source video
Basis decoder: a person is slicing bread
MARN: a man is spreading butter on a bread
Visual context in the memory
(c)
Source video
Basis decoder: a person is folding a piece of paper
MARN: a person is folding a paper airplane
Visual context in the memory
(d)

Figure 4. Qualitative comparison between the basis decoder and our MARN by examples from MSR-VTT and MSVD. For each example, we first show four representative images of the source video, then we show four context frames (corresponding to the 2D visual contexts in the memory) with high relevance (measured by attention weight) for the key words indicated by red color. MARN is able to correspond the video scene to key words due to the comprehensive understanding of the words by the designed memory scheme. Interestingly, the visual contexts between two adjacent key words in (c) and (d) are overlapped, which may help the model to learn the underlying association.

4.5.2 Comparison on MSVD

Similar to the experiments on MSR-VTT dataset, two groups of baselines are compared with our model on MSVD dataset: (1) fundamental methods including MP-LSTM with AlexNet as encoding scheme, S2VT and SA-LSTM that both use Inception-V4 for encoding, GRU-RCN [2] that leverages recurrent convolutional networks to learn video representation, HRNE [30] which proposes a Hierarchical Recurrent Neural Encoder to capture the temporal information of source videos, LSTM-E [32] which seeks to explore the decoding with LSTM and visual-semantic embedding simultaneously, LSTM-LS [26] which aims to model the relationships of different video-sequence pairs, h-RNN [62] that employs a paragraph generator to capture the inter-sentence dependency by sentence generators, aLSTMs [12] that models both encoder and decoder using LSTM with attention mechanism; (2) three newly published state-of-the-art methods, i.e., PickNet, RecNet and TSA-ED [56] which extracts the spatial-temporal representation in the trajectory level by structured attention mechanism.

The experimental results presented in Table 5 show that our MARN model performs significantly better than other methods on all metrics except BLEU-4. PickNet and RecNet achieve the best result on BLEU-4. Surprisingly, our basis decoder outperforms other methods substantially, which is mainly beneficial from our encoding scheme, i.e., the combination of 2D and 3D visual features in the specifically-designed way. The performance is further boosted by our Attended Memory Decoder.

Model	BLEU-4	METEROR	ROUGE-L	CIDEr
MP-LSTM (AlexNet) [47]	33.3	29.1	–	–
GRU-RCN [2]	43.3	31.6	–	68.0
HRNE [30]	43.8	33.1	–	–
LSTM-E [32]	45.3	31.0	–	–
LSTM-LS (VGG19+C3D) [26]	51.1	32.6	–	–
h-RNN [62]	49.9	32.6	–	65.8
S2VT (Inception-V4) [51]	39.6	31.2	67.5	66.7
aLSTMs [12]	50.8	33.3	–	74.8
SA-LSTM (Inception-V4) [51]	45.3	31.9	64.2	76.2
TSA-ED [56]	51.7	34.0	–	74.9
PickNet (V+L) [6]	**52.3**	33.3	69.6	76.5
RecNet$_{local}$(SA-LSTM) [51]	**52.3**	34.1	69.8	80.3
Basis decoder (ours)	47.5	34.4	71.4	89.9
MARN (ours)	48.6	**35.1**	**71.9**	**92.2**

Table 5. Performance of different video captioning models on MSVD dataset in terms of four metrics (%).

5. Conclusion

In this work, we have presented the Memory-Attended Recurrent Network (MARN) for video captioning. The model employs an attention-based recurrent network as the basis caption decoder and leverages a memory-based decoder to assist the decoding process. The memory is constructed to capture the full-spectrum correspondence between each candidate word and its various visual contexts across videos in training data, which enables the MARN to generate more precise captions for source videos . We show the superior performance of the MARN both quantitatively and qualitatively on two real-world datasets.

References

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and vqa. In *CVPR*, 2017.

[2] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. In *ICLR*, 2016.

[3] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop*, 2005.

[4] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu. Advances in optimizing recurrent networks. In *ICASSP*, 2013.

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

[6] Y. Chen, S. Wang, W. Zhang, and Q. Huang. Less is more: Picking informative frames for video captioning. In *ECCV*, 2018.

[7] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014.

[8] R. Das, M. Zaheer, S. Reddy, and A. McCallum. Question answering on knowledge bases and text using universal schema and memory networks. In *ACL (short paper)*, 2017.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[10] J. Dong, X. Li, W. Lan, Y. Huo, and C. G. Snoek. Early embedding and late reranking for video captioning. In *ACM Multimedia Conference*, 2016.

[11] Y. Feng, S. Zhang, A. Zhang, D. Wang, and A. Abel. Memory-augmented neural machine translation. In *EMNLP*, 2017.

[12] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen. Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055, 2017.

[13] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, 2013.

[14] G. Haffari and S. Maruf. Document context neural machine translation with memory networks. In *ACL*, 2018.

[15] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet. In *CVPR*, 2018.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] Y. Jia, Y. Ye, Y. Feng, Y. Lai, R. Yan, and D. Zhao. Modeling discourse cohesion for discourse parsing via memory network. In *ACL (Short Papers)*, volume 2, 2018.

[18] W. Jiang, L. Ma, Y.-G. Jiang, W. Liu, and T. Zhang. Recurrent fusion network for image captioning. In *ECCV*, 2018.

[19] Q. Jin, J. Chen, S. Chen, Y. Xiong, and A. Hauptmann. Describing videos using multi-modal fusion. In *ACM Multimedia Conference*, 2016.

[20] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 50(2), 2002.

[23] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. In *ICCV*, 2017.

[24] L. Li and B. Gong. End-to-end video captioning with multitask reinforcement learning. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019.

[25] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[26] Y. Liu, X. Li, and Z. Shi. Video captioning with listwise supervision. In *AAAI*, 2017.

[27] C. Ma, C. Shen, A. R. Dick, and A. van den Hengel. Visual question answering with memory-augmented networks. In *CVPR*, 2018.

[28] L. Ma, Z. Lu, and H. Li. Learning to answer questions from image using convolutional neural network. In *AAAI*, volume 3, 2016.

[29] M. Mohtarami, R. Baly, J. Glass, P. Nakov, L. Màrquez, and A. Moschitti. Automatic stance detection using end-to-end memory networks. In *NAACL-HLT*, 2018.

[30] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, 2016.

[31] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016.

[32] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016.

[33] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.

[34] V. Ramanishka, A. Das, D. H. Park, S. Venugopalan, L. A. Hendricks, M. Rohrbach, and K. . Multimodal video description. In *ACM Multimedia Conference*, 2016.

[35] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

[36] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *ICCV*, 2013.

[37] R. Shetty and J. Laaksonen. Frame-and segment-level features and candidate pool evaluation for video caption generation. In *ACM Multimedia Conference*, 2016.

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[39] J. Song, L. Gao, Z. Guo, W. Liu, D. Zhang, and H. T. Shen. Hierarchical lstm with adjusted temporal attention for video captioning. In *IJCAI*, 2017.

[40] S. Sukhbaatar, a. szlam, J. Weston, and R. Fergus. End-to-end memory networks. In *NIPS*. 2015.

[41] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *ICLR Workshop*, 2016.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[43] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[44] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015.

[45] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence - video to text. In *ICCV*, 2015.

[46] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *ICCV*, 2015.

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

[48] O. Vinyals, A. Toshev, S. Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[49] V. Voykinska, S. Azenkot, S. Wu, and G. Leshed. How blind people interact with visual content on social networking services. In *ACM Conference on Computer-Supported Cooperative Work & Social Computing*, 2016.

[50] B. Wang, L. Ma, W. Zhang, and W. Liu. Reconstruction network for video captioning. In *CVPR*, 2018.

[51] B. Wang, L. Ma, W. Zhang, and W. Liu. Reconstruction network for video captioning. In *CVPR*, 2018.

[52] J. Wang, W. Jiang, L. Ma, W. Liu, and Y. Xu. Bidirectional attentive fusion with context gating for dense video captioning. In *CVPR*, 2018.

[53] S. Wang, S. Mazumder, B. Liu, M. Zhou, and Y. Chang. Target-sensitive memory networks for aspect sentiment classification. In *ACL*, 2018.

[54] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, 2018.

[55] J. Weston, S. Chopra, and A. Bordes. Memory networks. In *ICLR*, 2015.

[56] X. Wu, G. Li, Q. Cao, Q. Ji, and L. Lin. Interpretable video captioning via trajectory structured localization. In *CVPR*, 2018.

[57] C. Xiao, J. Mei, and M. Müller. Memory-augmented monte carlo tree search. In *AAAI*, 2018.

[58] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[59] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. CVPR, June 2016.

[60] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *ICML*, 2015.

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

[62] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, 2016.

[63] Y. Yu, H. Ko, J. Choi, and G. Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, 2017.

[64] K.-H. Zeng, T.-H. Chen, C.-Y. Chuang, Y.-H. Liao, J. C. Niebles, and M. Sun. Leveraging video descriptions to learn video question answering. In *AAAI*, 2017.