Hierarchical Memory Modelling for Video Captioning

Junbo Wang[1,3], Wei Wang[1,3,*], Yan Huang[1,3], Liang Wang[1,2,3], Tieniu Tan[1,2,3]
[1]Center for Research on Intelligent Perception and Computing (CRIPAC),
National Laboratory of Pattern Recognition (NLPR), Beijing 100190, China
[2]Center for Excellence in Brain Science and Intelligence Technology (CEBSIT),
Institute of Automation, Chinese Academy of Sciences (CASIA)
[3]University of Chinese Academy of Sciences (UCAS)
{junbo.wang, wangwei, yhuang, wangliang, tnt}@nlpr.ia.ac.cn

ABSTRACT

Translating videos into natural language sentences has drawn much attention recently. The framework of combining visual attention with Long Short-Term Memory (LSTM) based text decoder has achieved much progress. However, the vision-language translation still remains unsolved due to the semantic gap and misalignment between video content and described semantic concept. In this paper, we propose a Hierarchical Memory Model (HMM) – a novel deep video captioning architecture which unifies a textual memory, a visual memory and an attribute memory in a hierarchical way. These memories can guide attention for efficient video representation extraction and semantic attribute selection in addition to modelling the long-term dependency for video sequence and sentences, respectively. Compared with traditional vision-based text decoder, the proposed attribute-based text decoder can largely reduce the semantic discrepancy between video and sentence. To prove the effectiveness of the proposed model, we perform extensive experiments on two public benchmark datasets: MSVD and MSR-VTT. Experiments show that our model not only can discover appropriate video representation and semantic attributes but also can achieve comparable or superior performances than state-of-the-art methods on these datasets.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**; **Hierarchical representations**;

KEYWORDS

Visual attention; hierarchical memory model; video captioning

∗Corresponding Author: Wei Wang.

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240538

ACM Reference Format:
Junbo Wang, Wei Wang, Yan Huang, Liang Wang, Tieniu Tan. 2018. Hierarchical Memory Modelling for Video Captioning. In *2018 ACM Multimedia Conference (MM'18), October 22-26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240538

1 INTRODUCTION

Automatically generating captions for videos is a challenging and important problem in computer vision. Along with the emergence of large-scale videos on the Internet, broadcasting channels, and other personal devices, video captioning can be applied to a wide range of applications, e.g., aiding visually impaired users, robotic vision, and incident report for surveillance.

Recently, deep encoder-decoder models which contain an attention-based video encoder and a LSTM-based text decoder have achieved encouraging performance in video captioning [1, 14, 39, 42]. However, the semantic gap and misalignment between video content and described semantic concept are still the fundamental problems in video captioning. Recently, selective attribute based methods [41] have been proposed to solve these problems and achieved much better results. They generally focus on the semantic attribute attention based on previous words, which overlook the effects of the long-term textual and visual contents. As we know, neural memory models, e.g., neural turing machines [10] and memory networks [23], have been successfully applied to long-term sequence modelling, e.g., visual question answering [35]. It is natural to apply memory networks to model the long-term textual and visual contents which helps to select semantic attributes in video captioning.

In this paper, we propose a hierarchical memory model to describe videos in Figure 1 (a), which consists of a textual memory, a visual memory and an attribute memory. The attribute memory is built on the textual memory and visual memory in a hierarchical way. The textual memory and visual memory guide attention for semantic attribute selection in addition to modelling the long-term dependency for video sequences and sentences. Considering that attributes play a very important role in the sentence generation, we also propose an improved video attribute detection method. Figure 1 (b) shows an example of detected attributes and generated

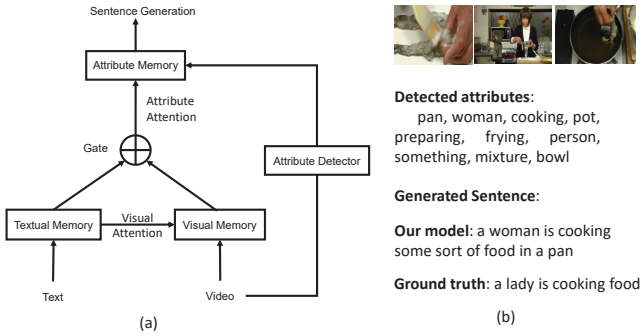


Figure 1: (a) The illustration of our hierarchical memory model. It consists of a textual memory, a visual memory and an attribute memory. The attribute memory is built on the textual memory and visual memory. The textual memory and visual memory guide attention for semantic attribute selection. We also propose an improved video attribute detection method. (b) An example of detected attributes and generated sentence for a video.

sentence for a video. It can be seen that the detected attributes help our model generate a sentence containing more details.

Inspired by Neural Turing Machine (NTM) [10] and Memory Networks [23], we utilize three external memories in terms of textual memory, visual memory, and attribute memory to store the hidden representations of LSTM-based text decoder, the attended video representations, and the semantic attributes, respectively. Take an input video for example, we first extract its frame/clip representations with pretrained 2D/3D deep networks (e.g., VGG), and extract video attributes by an improved attribute detector. Then the proposed hierarchical memory model performs as follows: 1) the LSTM-based text decoder writes its hidden states into the textual memory (TM) after predicting the next word, 2) the visual attention model exploits textual information read from the updated TM to select local video representations, which will be written into the visual memory (VM), 3) combine the contents read from VM and TM to select relevant semantic attributes from an attribute memory (AM), 4) the selected attributes are fed into the LSTM-based text decoder to generate words.

The main contributions of this paper are summarized as follows:

- We are the first to propose a hierarchical memory model for video captioning, which contains not only a textual memory and a visual memory, but also an attribute memory for incorporating semantic attributes.
- We propose an improved video attribute detection framework, which overcomes the learning difficulty of previous multiple instance learning methods.

2 RELATED WORK

Video Captioning Most methods in this direction can be divided into two categories: language template based methods [11, 20, 25] and neural network based methods [14, 29, 30, 32, 38, 39, 42]. The language template based methods first detect key semantic words from visual content, and then relate them to produce the sentence with predefined language template (e.g., a syntactically well-formed tree). Accordingly, this kind of methods usually generate grammatically correct sentences, but undermine the novelty and flexibility of the sentence. The neural network based methods inspired by the success of Neural Machine Translation consider video captioning as a result of translating videos to sentences. Venugopalan et al. [30] extend the image captioning methods by simply applying the mean pooling to the representations of all frames. However, this method obviously breaks the spatiotemporal structure information of a video, and has many limitations in describing videos with great temporal dynamics. To deal with the issue, some researchers [7, 29, 36] use recurrent neural networks (RNNs) to model temporal dependencies in videos. Yao et al. [39] propose a temporal attention mechanism to exploit global temporal structure in videos. Some others [1, 14] present a more sophisticated recurrent encoder to exploit spatiotemporal information of videos. The proposed method in this paper belongs to the second category.

Memory Modelling Vanilla recurrent neural networks (RNN) whose recurrent hidden states can be viewed as an implicit memory has difficulty in modelling long-range temporal dependencies [3]. Long short-term memory (LSTM) and gated recurrent units (GRU) have been proposed to improve RNN. However, all these models are still limited in modelling long-range temporal dependencies. To deal with it, several works on memory modelling have been proposed, which has proceeded along two different directions: neural turing machine and memory networks. Neural Turing Machine (NTM) proposed by Graves et al. [10] holds an external memory to interact with the internal state of neural networks via selective read and write operations, and it has shown great potential in storing and modifying the internal state of the network over long time periods. Different form NTM, memory networks [33] adopt static external memory which makes it easy to learn in real tasks. Several variants of memory networks have been successfully applied to textual question answering [43] and visual question answering [35]. In this work, we devise three external memories to store different modal information in a hierarchical way, which improve the performance of video captioning.

Semantic Attributes in Sequence Learning Following the fact that semantic attributes detected from visual content have been widely used in visual recognition [18], several recent works have applied semantic attributes into sequence learning for image/video captioning. Fang et al. [9] first use multiple instance learning to train visual detectors for semantic attributes (e.g., nouns, verbs, and adjectives), then use a maximum-entropy language model to generate sentences based on the outputs of visual detectors, finally

re-rank sentence candidates via a deep multimodal similarity model. Along with this recipe, Tran et al. [27] design a larger entity recognition model which detects celebrities and landmarks to improve the caption, You et al. [41] learn to selectively attend semantic attributes via parametric and non-parametric attribute prediction models. Later Wu et al. [34] incorporate high-level concepts into the successful CNN-RNN model which can improve the performance of vision-to-language problems. At the same time, [16, 40] also demonstrate the conclusion by injecting attributes into the encoder-decoder model in image/video captioning via different ways. In this paper, we propose an improved video attribute detector which overcomes the learning difficulty of previous multiple instance learning methods.

3 MAIN METHOD

In this section, we will first describe the overall framework of the proposed HMM, and then introduce four key components: video attribute detection, LSTM with semantic attributes, memory-augmented attention, and hierarchical memories. Finally, we explain the details of model learning.

3.1 Overall Framework

The overall framework of the proposed HMM is illustrated in Figure 2. For a given video V, we sample N_v equally-spaced frames from the video, and employ pretrained 2D/3D Convolutional Neural Networks (CNNs) to extract frame/clip features $\{v_1, v_2, v_3, \ldots, v_n\}$, where n is the number of sampled frames/clips. After the video representation learning, the framework can be divided into four key parts. (1) **Video attribute detection**. We pretrain a video attribute detection model to obtain the attributes $\{a_i, i = 1, 2, \cdots, K\}$ that occur in the video. These attributes are stored in the attribute memory (AM). (2) **LSTM with semantic attributes**. At each time t, the LSTM based text decoder takes the input word y_t, previous hidden state h_{t-1}, and the attributes read from AM as input, and output current hidden states. (3) **Memory-augmented attention**. The memory-augmented attention (e.g., $Attend^t$) is designed to adaptively select relevant visual contents or video attributes based on the current memory status (TM^t or VM^t). (4) **Hierarchical memories**. When the hidden state h_t of LSTM evolves over time, it will be first written into the textual memory TM^t, which updates textual memory to TM^{t+1}. Then the updated textual memory contents will be read out to guide attention model to select relevant visual information $Attend^{t+1}$, which will be written into the visual memory VM^{t+1}. Combining the updated textual memory TM^{t+1} and visual memory VM^{t+1}, video attributes can be selected from the attribute memory via an adaptive gate. Finally, the selected attributes will be injected into the LSTM-based text decoder to predict the next word.

3.2 Video Attribute Detection

To detect video attributes, a common way is to train a multiple instance learning [9] model on the video frames.

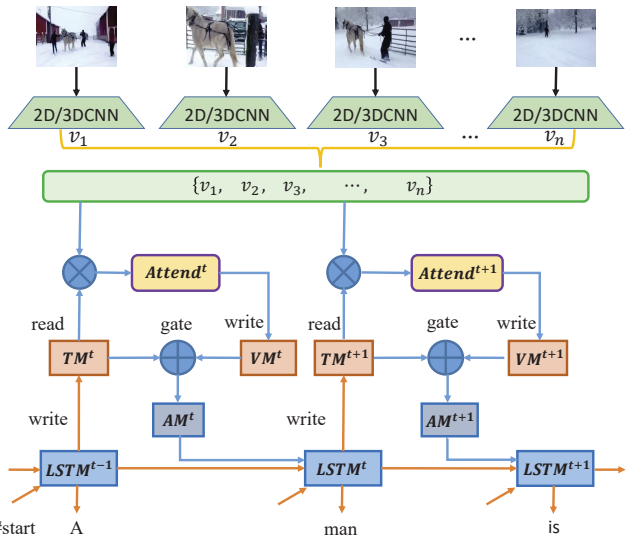


Figure 2: The overall framework of the proposed hierarchical memory model for video captioning. It contains three memories: textural memory (TM), visual memory (VM) and attribute memory (AM). Particularly, the AM stores video attributes by our improved attribute detector. The gate unit controls how to use visual information and textual information to select attributes.

However, this method will lead to the semantics noise [16] due to simply assigning video description to each sampled frame during the process of model training. To address this problem, Pan et al. [16] propose an improved video MIL model. For each attribute $w_a \in A$, all the spatial regions of the sampled N_v frames in a video are regarded as a bag. The bag is considered as positive if w_a is in the video V's descriptions and negative otherwise. When using a noisy-OR version of MIL [31], the probability of bag b_V containing attribute w_a is measured on the probabilities of all instances in this bag, which can be calculated as:

$$P_V^{w_a} = 1 - \prod_{j \in \{1, N_v\}} \prod_{i \in b_V^j} \left(1 - p_{ij}^{w_a}\right) \quad (1)$$

where $p_{ij}^{w_a}$ denotes the probability of the attribute w_a that the i-th region in the j-th frame predicts and b_V^j denotes all sub-regions corresponding to the j-th frame in the video V. Actually, the number of multiplication of $\left(1 - p_{ij}^{w_a}\right)$ is usually very large and exceeds the lower limit of floating point arithmetic precision. To avoid this problem, we propose an improved schemes as illustrated in Figure 3. The proposed scheme in Figure 3 takes the sampled N_v frames as input, and orderly forwards the input through FCN layers, a sigmoid layer, an image MIL layer and a max layer. The FCN architecture is a fully convolutional neural network transformed from the VGG-16 [22]. The image MIL layer performs multiple instance learning on the output response map produced by

the FCN layer. The procedure can be formulated as:

$$P_j^{w_a} = 1 - \prod_{i \in b_V^j} (1 - p_i^{w_a}) \quad (2)$$

where $p_i^{w_a}$ is the probability generated by the i-th region of the response map, b_V^j denotes the j-th frame in the video V, and $P_j^{w_a}$ denotes the probability of j-th frame containing attribute w_a. After that, the max layer computes the final probability by taking the maximum value along the frame sampling dimension. Similar to previous MIL model [9], a cross entropy loss is employed during the process of model training.

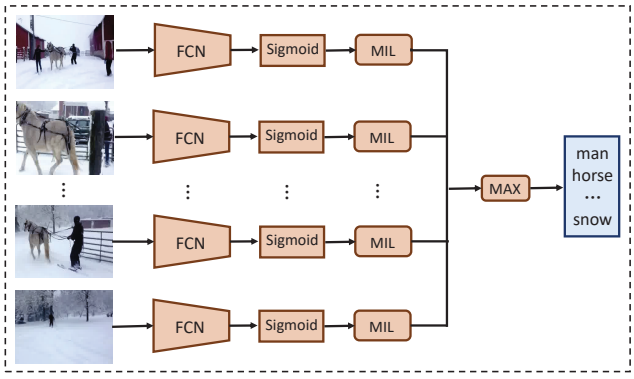


Figure 3: The proposed video attribute detection architecture for the video frames. In particular, FCN denotes a Fully Connected Neural Network transformed from VGG Net.

3.3 LSTM with Semantic Attributes

Different from the widely used unimodal LSTM [44], we take the attended semantic attributes as another input. Here the attributes a_t are read from our attribute memory during caption generation. For each word in the sentence, we denote it as a vector y_t via one-hot encoding, and then transform it to an embedding vector E_t. The whole procedure can be formulated as follows:

$$i_t = \sigma\left(W_i E_t + U_i h_{t-1} + M_i a_t + b_i\right) \quad (3)$$

$$f_t = \sigma\left(W_f E_t + U_f h_{t-1} + M_f a_t + b_f\right) \quad (4)$$

$$o_t = \sigma\left(W_o E_t + U_o h_{t-1} + M_o a_t + b_o\right) \quad (5)$$

$$\tilde{c}_t = \phi\left(W_c E_t + U_c h_{t-1} + M_c a_t + b_c\right) \quad (6)$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \quad (7)$$

$$h_t = o_t \odot \phi\left(c_t\right) \quad (8)$$

where the default operation between matrices is matrix multiplication, $\odot$ is an element-wise multiplication, W, U, M, and b are the parameters to be learned, σ is the element-wise logistic sigmoid function, and ϕ is hyperbolic tangent function tanh.

For clear illustration, we abbreviate the above process as:

$$h_t = \psi\left(h_{t-1}, c_{t-1}, y_t, a_t\right) \quad (9)$$

3.4 Memory-augmented Attention

Given the visual features $V = \{v_1, v_2, \ldots, v_n\}$, and the r_{tm}^t read from the textual memory at time t, we feed them through a fully-connected network followed by a softmax function to compute the attended weight α_t^i:

$$\alpha_t^i = softmax(w_h^T \tanh\left(W_r r_{tm}^t + U_\alpha v_i + b_\alpha\right)) \quad (10)$$

where w_h^T, W_r, U_α, and b_α denote the parameters to be learned. With the attention distribution α_t^i, the attended visual representation at time t can be computed as:

$$c_t = \sum_{i=1}^{n} \alpha_t^i v_i \quad (11)$$

where c_t will be written into the visual memory as presented below. To simplify description, the above attention model can be abbreviated as follows:

$$c_t = \beta(V, r_{tm}^t) \quad (12)$$

The attention model for selecting semantic attributes is similar to β, except that the inputs are different.

3.5 Hierarchical Memories

The proposed hierarchical memory model (HMM) consists of three external memories: textual memory (TM), visual memory (VM) and attribute memory (AM). TM and VM are implemented as a matrix $M \in R^{N \times D}$, respectively, where N denotes the number of memory locations and D denotes the vector size of each location. AM is a similar matrix except that each row stores the embedding representation of an attribute. Since the semantic attributes are very high-level representations for the video captioning, we keep the AM static during sentence generation. During the interaction between controller (e.g., the LSTM and the visual attention model) and memory, we employ the similar read/write operation and content-based addressing mechanism detailed in [10]. To facilitate the following descriptions, the read/write operation can be defined as:

$$r_t = f_{read}\left(w_t, M_t\right) \quad (13)$$

$$M_t = f_{write}\left(u_t, e_t, a_t, M_{t-1}\right) \quad (14)$$

Writing hidden representations to textual memory After predicting the next word, the controller (the LSTM-based text decoder) will write the hidden representations into the textual memory TM^t to store textual information. The address vector u_t^{tm}, the erase vector e_t^{tm} and the add vector a_t^{tm} are emitted by the write head of the controller, respectively. The textual memory is updated by:

$$TM^t = f_{write}\left(u_t^{tm}, e_t^{tm}, a_t^{tm}, TM^{t-1}\right) \quad (15)$$

Reading from the updated textual memory The address vector w_t^{tm} is emitted by the read head of the controller (the attention model). Once the textual memory is updated, the read vector returned by the controller is computed by:

$$r_t^{tm} = f_{read}\left(w_t^{tm}, TM^t\right) \quad (16)$$

Attention selection for video representations After the read vector r_t^{tm} is gained from the textual memory, the

attended visual information at current time for the video representations $V = \{v_1, v_2, v_3, \ldots, v_n\}$ can be obtained by:

$$c_t = \beta(V, r_t^{tm}) \tag{17}$$

Writing attended visual information to visual memory After obtaining the attended visual information, the controller (the attention model) will write the visual information into the visual memory VM^t. Similarly, the address vector u_t^{vm}, the erase vector e_t^{vm} and the add vector a_t^{vm} are emitted by the write head of the controller, respectively. The visual memory is updated by:

$$VM^t = f_{write}\left(u_t^{vm}, e_t^{vm}, a_t^{vm}, VM^{t-1}\right) \tag{18}$$

Reading from the updated visual memory The w_t^{vm} is emitted by the read head of the controller (the attention model). Once the visual memory is updated, the read vector from the visual memory is written as:

$$r_t^{vm} = f_{read}\left(w_t^{vm}, VM^t\right) \tag{19}$$

Gated semantic attention for attribute memory Since the generation of next word relies on either visual clues or textual clues [13], we devise a novel gate to decide the optimal proportion between visual clues and textual clues. Based on the hidden representation of the LSTM-based text decoder and the external textual memory status, the adjusted gate s_t is defined as:

$$s_t = \sigma\left(W_s r_t^{tm} + U_s h_{t-1} + b_s\right) \tag{20}$$

where W_s, U_s, and b_s denote the parameters to be learned, and σ denotes the sigmoid activation function. Given the adjusted gate s_t, the visual clues r_t^{vm}, the textual clues r_t^{tm} and the semantic attributes $A = \{a_1, a_2, a_3, \ldots, a_K\}$, the attended attributes are obtained by:

$$a_t = \beta(g\left(A\right), g\left(s_t r_t^{tm} + \left(1 - s_t\right) r_t^{vm}\right)) \tag{21}$$

where g denotes the activation function Relu and β denotes the proposed memory-augmented attention model.

Computing hidden representations of LSTM After relevant attributes are read from the attribute memory, the hidden representations of LSTM-based text decoder are updated by:

$$h_t = \psi\left(h_{t-1}, c_{t-1}, y_{t-1}, a_t\right) \tag{22}$$

The updated hidden representation h_t will be used for predicting next word.

3.6 Model Learning

During the training phase, the model can be learned by minimizing the following objective function on training video-description pairs $\left\{\left(x^i, y^i\right) | i = 1, 2, \cdots, M\right\}$:

$$L\left(\theta\right) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{T_i}\log\rho\left(y_j^i | y_{1:j-1}^i, x^i, \theta\right) + \lambda\left\|\theta\right\|_2^2 \tag{23}$$

where x^i denotes the input video, y^i denotes the corresponding sentence whose length is T_i, θ denotes all the parameters to be learned, and λ denotes the regularization term coefficient. We use a stochastic gradient descent algorithm with an adaptive learning rate to learn the above model parameters.

The probability distribution over the whole vocabulary at time t can be formulated as:

$$z_t = tanh\left(W_v v_t + W_h h_t + b_h\right) \tag{24}$$

$$\rho_t = softmax\left(U_\rho z_t + b_\rho\right) \tag{25}$$

where W_v, W_h, b_h, U_ρ, and b_ρ denote the parameters to be learned. Once the probability distribution ρ_t is determined, each word y_t can be sampled from the vocabulary until the emergence of the end tag of sentence.

4 EXPERIMENTS

We perform experiments on two public datasets: MSVD [4] and MSR-VTT [37] to demonstrate the effectiveness of the proposed model. We will first describe the experiment datasets and settings, and then make comparisons with the state-of-the-art methods.

4.1 Datasets

MSVD This dataset is composed of 1970 short videos collected from YouTube. Each video is annotated with about 40 English sentences, and the total number of video-description pairs in the dataset is 80839. In our experiment, we adopt the standard split used in [39], where 1200 videos, 100 videos, and 670 videos are used for training, validation, and testing, respectively.

MSR-VTT This dataset is a recent collection of large-scale video description dataset with the largest number of clip-sentence pairs and word vocabulary. It consists of 10000 short videos, and each video clip is equipped with about 20 English sentences. Following the default split in [37], where 6513 videos, 497 videos, and 2990 videos are used for training, validation, and testing, respectively.

4.2 Experimental Settings

In our experiments, we uniformly sample 30 frames and 40 frames from each video in the MSVD and MSR-VTT datasets, respectively. For video representations, we extract the output of the pool3 layer from Inception-V3 [24], the last fully connected layer from VGG19 [22] and the fc6 layer from C3D [26] for frame and clip representation, respectively, and concatenate them along the last dimension of features to form the final video representation. For semantic attribute detection in the video, we first choose 1,000 most common words on each dataset as the candidate semantic attributes. Then, we train the proposed video attribute detector on each training data, and obtain a 1,000-way vector of probabilities corresponding to the 1,000 attributes. Finally, we sort the attribute probabilities and select the top 20 attributes as the final attributes. For text representation of each sentence, we apply the common lowercasing and rare word elimination to all the descriptions and use one-hot encoding to represent each word. For the training model, the hidden size of a single layer LSTM is 512, the word embedding dimension is 468, the sizes of both textual memory and visual memory are (128,512), the batch size is 64, and the beam size is 5. To address variable-length sentences, a start tag and a end tag

Method	BLEU@4	METEOR	CIDEr-D
FGM [25]	13.68%	23.90%	-
LSTM [30]	33.29%	29.07%	-
SA [39]	41.92%	29.60%	51.67 %
S2VT [29]	-	29.8%	-
LSTM-E [15]	45.3%	31.0%	-
p-RNN [42]	49.9%	32.6%	65.8 %
HRNE [14]	43.8%	33.1%	-
BGRCN [1]	48.42%	31.70%	65.38
HBA [2]	42.5%	32.4%	63.5
RMA [12]	45.7%	31.9%	57.3
TSA [16]	52.8%	33.5%	74.0
HMM-c3d	46.0%	31.8%	63.2%
HMM-vgg	48.7%	32.4%	67.6%
HMM-inv3	49.9%	33.0%	70.7%
HMM-c3d-vgg	50.8%	33.1%	72.2%
HMM-c3d-inv3	**52.9%**	**33.8%**	**74.5%**

Table 1: The evaluation performance on BLEU@4, METEOR and CIDEr-D metrics compared with the other eight state-of-the-art methods on MSVD.

Method	BLEU@4	METEOR	CIDEr-D
EMLR [8]	38.7%	26.9%	**45.9**
MVD [19]	38.3%	27.0%	41.8
Aalto [21]	39.8%	26.9%	45.7
MP-vgg [30]	34.8%	24.8%	-
SA-vgg [39]	35.6%	25.4%	-
SA-c3d [39]	36.1%	25.7%	-
SA-c3d-vgg [39]	36.6%	25.9%	-
SA-inv3 [39]	36.3%	23.6%	37.7%
SA-c3d-inv3 [39]	38.1%	25.9%	38.0%
HMM-c3d	37.4%	26.2%	40.5%
HMM-vgg	36.8%	26.0%	38.6%
HMM-inv3	37.2%	26.5%	39.4%
HMM-c3d-inv3	**39.9%**	**28.3%**	40.9%

Table 2: The evaluation performance on BLEU@4, METEOR and CIDEr-D metrics compared with recent state-of-the-art methods on MSR-VTT.

are added to sentence. To avoid the interruption of redundant sentences, the sentences with length larger than 30 in the dataset are removed from the datasets. To prevent gradient explosion, we clip the gradients to the range of $(-10, 10)$. We adopt the Adadelta optimizer with a learning rate of 1e-4 in the training stage. To compare with the state-of-the-art methods fairly, we employ the three common evaluation metrics, i.e., BLEU@4 [17], METEOR [6] and CIDEr-D [28], and use the codes published by Microsoft COCO Evaluation Server [5] to compute all evaluation metrics.

4.3 Quantitative Analysis

4.3.1 Performance on MSVD. We compare our model on MSVD with several state-of-the-art models: FGM [25], LSTM [30], SA [39], S2VT [29], LSTM-E [15], p-RNN [42], HRNE [14], BGRCN [1], HBA [2], RMA [12] and TSA [16]. To make a fair comparison with these methods, we report the results on all the individual feature: Inception-V3, VGG, C3D and the combination of them. Table 1 shows the evaluation results of different models on MSVD in terms of BLEU@4, METEOR, and CIDEr-D. To be noted, most of the state-of-the-art results are produced in the combination of VGG and C3D. We can see that our proposed HMM achieves comparable or better performance than the state-of-the-art methods in almost all metrics. Specifically, our HMM performs better than SA by $\frac{52.9-41.92}{41.92} = 26.2\%$ in the BLEU@4 score, by $\frac{33.8-29.6}{29.6} = 14.2\%$ in the METEOR score, and by $\frac{74.5-51.67}{51.67} = 44.2\%$ in the CIDEr-D score, respectively. Compared with p-RNN [42], our proposed HMM also outperforms it by $\frac{72.2-65.8}{65.8} = 9.7\%$ in the CIDEr-D score when using the same feature (VGG+C3D). Since SA [39], p-RNN [42], BGRCN [1], and HRNE [14] are representative attention-based methods, which devise some complicated video encoders to get better representations, the comparison with these methods indicates exploring more effective attention and memory mechanisms are very important in video captioning. Even compared with RMA [12] using a different key-value memory network, our proposed HMM performs much better than it by a large margin in all evaluation metrics. Furthermore, our proposed HMM can achieve comparable performance than the best competitor TSA [16] while TSA [16] employs the best-performing architecture (factored, two-layer LSTM) in [7]. These results further demonstrate the effectiveness of our HMM in describing natural videos.

4.3.2 Performance on MSR-VTT. The performance comparisons to recent state-of-the-art methods on MSR-VTT are shown in Table 2. MSR-VTT is a newly released large-scale video benchmark [37], which is very challenging for video captioning due to the largest number of video-sentence pairs. In this experiment, we cite recent state-of-the-art results reported on the MSR-VTT dataset, e.g., EMLR, MVD, Aalto, MP-vgg, SA-vgg, SA-c3d and SA-c3d-vgg, and reimplement SA-inv3 and SA-c3d-inv3. All the results of these models in Table 2 are conducted on the individual feature: Inception-V3, VGG, C3D and the combination of them. From these results, we can see that the proposed HMM-inv3 performs better than the baseline method SA-inv3 in terms of all three evaluation metrics. When combined with C3D, the HMM-c3d-inv3 improves the performance to 39.9% in the BLEU@4, 28.3% in the METEOR, and 40.9% in the CIDEr-D, respectively. The performance improvement further demonstrates the effectiveness of the proposed model.

4.4 Qualitative Analysis

To better understand the proposed HMM, we first visualize some generated sentences and attributes on the test set of MSVD in Fig. 4. From these generated results, we can see that

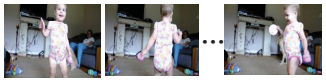
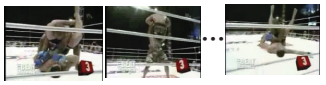
	Attributes from Video: Ball, teams, players, game, people, men, player, soccer, football, playing	Generated Sentence: SA: people are playing HMM: a group of men are playing soccer	Reference Sentence: 1. two teams are playing soccer 2. men are playing football 3. some men are playing soccer
	Attributes from Video: playing, going, person, moving, man, dog, front, funny, white, cute	Generated Sentence: SA: a man is playing a guitar HMM: a man is playing with a dog	Reference Sentence: 1. a man is petting two dogs 2. a man pets some dogs 3. a man is play with pets
	Attributes from video: child, little, baby, kid, toddler, cute, playing, using, trying, high	Generated Sentence: SA: a baby is playing on a couch HMM: a baby is playing with a toy	Reference Sentence: 1. a baby is playing with toys 2. a toddler is picking up toys 3. the baby is putting away her toys
	Attributes from video: pan, woman, cooking, pot, preparing, frying, person, something, mixture, bowl	Generated Sentence: SA: a woman is cooking the kitchen HMM: a woman is cooking some sort of food in a pan	Reference Sentence: 1. a lady is cooking food 2. the lady prepare the food 3. a person making nabeyaki udon noodle
	Attributes from video: Man, his, playing, animal, boxing, lady, black, person, around, doing	Generated Sentence: SA: the monkey is playing a monkey HMM: two men are boxing in a boxing ring	Reference Sentence: 1. two men are boxing in a ring 2. two men fight inside a ring 3. two persons are fighting

Figure 4: Descriptions and Attributes generated on the test set of MSVD. The attributes (only show top 10 words) are generated by our proposed video attribute detector. The generated sentences are from the baseline method SA [39] and our proposed HMM, respectively. We also show the human-annotated reference sentences.

our proposed HMM can generate more accurate descriptions compared with the baseline method SA [39]. For example, the generated word "dog " by HMM is much better than the word "guitar " generated by SA for the second video. Moreover, our proposed HMM can capture more details about video content than SA. For example, HMM can generate the fine-grained words "a group of " and "soccer " while the generated sentence by SA is ambiguous in the first video. It should be noted that our hierarchical memories provide a more compact interactions for the attention of visual features and semantic attributes. Even the video attribute detector does not detect the attribute word "toy ", our HMM can also produce the key word "toy " compared with SA. We further visualize the attention weight shift for temporal visual features and semantic attributes on sampled frames in a test video in Fig. 5. We can see that the attention shift of each word in the generated sentence is much consistent to the video contents and attributes, which demonstrates that the proposed model can enhance the attention and memory modeling for video captioning.

4.5 Model Analysis

To figure out how these additional memory modules contribute to our proposed HMM, we design some incremental experiments on it. Table 3 shows the performance of six different proposed model variants on the MSVD dataset using the VGG19 feature. The first row shows the results of the baseline of the proposed HMM which does not contains the textual memory, the visual memory and the attribute memory. The second line to the sixth line shows the results of the proposed HMM which contains part or all of the three memory modules. In particular, the HMM (only-T) and HMM (only-V) both performs better than the baseline HMM (no-T-V-A), especially in the CIDEr-D metric. Combining these two memory modules, the HMM (only-T-V) achieves 47.5% in the BLEU@4 metric, making the improvement over the baseline HMM (no-T-V-A) by $\frac{47.5-45.9}{45.9} = 3.5\%$, which proves the effectiveness of additional textual memory and visual memory. Similarly, the HMM (only-A) can achieve much better performance than the baseline HMM (no-T-V-A), which suggests that semantic attributes are very important to video captioning. When adding all these memory modules together, the HMM (T-V-A) can further obtain the best performance in these model variants in terms of all evaluation metrics. These results demonstrate the effectiveness of each additional memory module in the task.

5 CONCLUSIONS

In this paper, we have proposed a hierarchial memory model which enhances video feature and attribute attention for attribute-based video captioning. To exploit more rich semantic attributes for videos, we improve the previous MIL framework in terms of network architecture. To incorporate the semantic attributes into our system, we attach the LSTM-based text decoder and the attention model for video presentations with textual memory and visual memory respectively

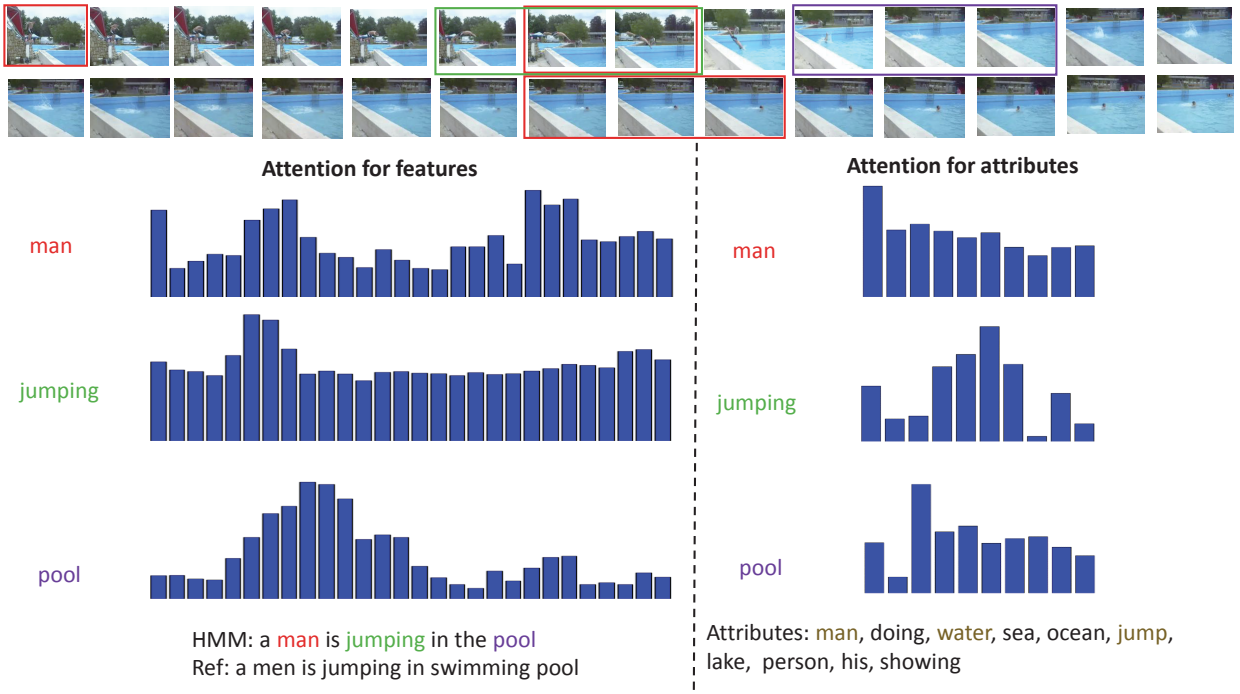


Figure 5: Visualization of generated sentences, attributes and corresponding attention shift about a video on the test of MSVD. The top displays sampled 28 frames from the video, the left shows the attention shift on the sampled frames with regard to each key word in the generated sentence, the right shows the attention shift on the top 10 attributes stored in the attribute memory about each key word in the generated sentence.

Method	BLEU@4	METEOR	CIDEr-D
HMM (no-T-V-A)	45.9%	30.5%	61.8%
HMM (only-T)	46.8%	30.8%	63.6%
HMM (only-V)	46.2%	31.2%	63.5%
HMM (only-T-V)	47.5%	31.6%	65.7%
HMM (only-A)	47.1%	31.8%	64.4%
HMM (T-V-A)	48.7%	32.4%	67.6%

Table 3: The performance comparison on the MSVD dataset of six different model variants with the VG-G19 feature in terms of three evaluation metrics. Here T, V and A denotes Textual Memory, Visual Memory and Attribute Memory, respectively.

to guide proper visual information flow into the selection of semantic attributes. The performance comparisons with other state-of-the-art methods on two publicly benchmark datasets demonstrate the effectiveness of our model.

ACKNOWLEDGMENTS

This work is jointly supported by National Key Research and Development Program of China (2016YFB1001000), National Natural Science Foundation of China (61420106015,61572504). In addition, this work is also supported by Huawei Innovation Research Program (HIRP) and grants from NVIDIA and the NVIDIA DGX-1 AI Supercomputer.

REFERENCES

[1] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. 2016. Delving Deeper into Convolutional Networks for Learning Video Representations. *Proceedings of the International Conference on Learning Representations* (2016).

[2] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. 2017. Hierarchical boundary-aware neural encoder for video captioning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.

[4] David L Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

[5] 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* (2015).

[6] Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

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

[8] Jianfeng Dong, Xirong Li, Weiyu Lan, Yujia Huo, and Cees G-M Snoek. 2016. Early embedding and late reranking for video captioning. In *Proceedings of the 2016 ACM on Multimedia*

Conference. ACM, 1082–1086.
[9] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. 2015. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[10] Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv* (2014).
[11] Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond J Mooney, Kate Saenko, and Sergio Guadarrama. 2013. Generating Natural-Language Video Descriptions Using Text-Mined Knowledge. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*.
[12] Arnav Kumar Jain, Abhinav Agarwalla, Kumar Krishna Agrawal, and Pabitra Mitra. 2017. Recurrent Memory Addressing for Describing Videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[13] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2016. Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning. *arXiv* (2016).
[14] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. 2016. Hierarchical Recurrent Neural Encoder for Video Representation with Application to Captioning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016).
[15] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2016. Jointly modeling embedding and translation to bridge video and language. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016).
[16] Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. 2017. Video Captioning with Transferred Semantic Attributes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017).
[17] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.
[18] Devi Parikh and Kristen Grauman. 2011. Relative attributes. In *Proceedings of the International Conference on Computer Vision*.
[19] Vasili Ramanishka, Abir Das, Dong Huk Park, Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, and Kate Saenko. 2016. Multimodal video description. In *Proceedings of the 2016 ACM on Multimedia Conference*. 1092–1096.
[20] Marcus Rohrbach, Wei Qiu, Ivan Titov, Stefan Thater, Manfred Pinkal, and Bernt Schiele. 2013. Translating video content to natural language descriptions. In *Proceedings of the International Conference on Computer Vision*.
[21] Rakshith Shetty and Jorma Laaksonen. 2016. Frame-and segment-level features and candidate pool evaluation for video caption generation. In *Proceedings of the 2016 ACM on Multimedia Conference*. 1073–1076.
[22] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv* (2014).
[23] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Proceedings of the Advances in Neural Information Processing Systems*.
[24] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision. *arXiv* (2015).
[25] Jesse Thomason, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Raymond J Mooney. 2014. Integrating Language and Vision to Generate Natural Language Descriptions of Videos in the Wild. In *Proceedings of the 25th International Conference on Computational Linguistics*.
[26] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the International Conference on Computer Vision*.
[27] Kenneth Tran, Xiaodong He, Lei Zhang, Jian Sun, Cornelia Carapcea, Chris Thrasher, Chris Buehler, and Chris Sienkiewicz. 2016. Rich image captioning in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[28] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[29] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence-video to text. In *Proceedings of the International Conference on Computer Vision*.
[30] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2014. Translating videos to natural language using deep recurrent neural networks. *arXiv* (2014).
[31] Paul Viola, John C Platt, Cha Zhang, et al. 2005. Multiple instance boosting for object detection. In *Proceedings of the Advances in Neural Information Processing Systems*.
[32] Junbo Wang, Wei Wang, Yan Huang, Liang Wang, and Tieniu Tan. 2018. M3: Multimodal Memory Modelling for Video Captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7512–7520.
[33] Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory Networks. In *Proceedings of the International Conference on Learning Representations*.
[34] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton van den Hengel. 2016. What value do explicit high level concepts have in vision to language problems?. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[35] Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *Proceedings of the International Conference on Machine Learning*. 2397–2406.
[36] Huijuan Xu, Subhashini Venugopalan, Vasili Ramanishka, Marcus Rohrbach, and Kate Saenko. 2015. A multi-scale multiple instance video description network. *Proceedings of the International Conference on Computer Vision* (2015).
[37] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[38] Jun Xu, Ting Yao, Yongdong Zhang, and Tao Mei. 2017. Learning Multimodal Attention LSTM Networks for Video Captioning. In *Proceedings of the 2017 ACM on Multimedia Conference*. 537–545.
[39] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Describing videos by exploiting temporal structure. In *Proceedings of the International Conference on Computer Vision*.
[40] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. 2016. Boosting image captioning with attributes. *arXiv* (2016).
[41] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
[42] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. 2016. Video paragraph captioning using hierarchical recurrent neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016).
[43] Wojciech Zaremba, Tomas Mikolov, Armand Joulin, and Rob Fergus. 2016. Learning simple algorithms from examples. In *Proceedings of the International Conference on Machine Learning*. 421–429.
[44] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv* (2014).