Pattern Recognition Letters 105 (2018) 23–29

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Multimodal architecture for video captioning with memory networks and an attention mechanism

Wei Li*, Dashan Guo, Xiangzhong Fang

Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China

ARTICLE INFO

Article history:
Available online 12 October 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Video captioning
Memory network
Attention mechanism

ABSTRACT

Automatically describing videos containing rich and open-domain activities is a very challenging task for computer vision and machine learning research. Obviously, accurate descriptions of video contents need the understanding of both visual concepts and their temporal dynamics. A lot of efforts have been made to understand visual concepts in still image tasks, e.g., image classification and object detection. However, the combination of visual concepts and temporal dynamics has not been given sufficient attention. To delve deeper into the unique characteristic of videos, we propose a novel video captioning architecture to integrate both visual concepts and temporal dynamics. In this paper, an attention mechanism and memory networks are combined together into our multimodal framework with a feature selection algorithm. Specially, we utilize the soft attention mechanism to choose visual concepts relevant frames based on previously generated words, and the memorization of temporal dynamics is implemented by the memory networks, which have great advantages of memorizing long-term information. Then the visual concepts and the temporal dynamics are integrated together into our multimodal architecture. Moreover, the feature selection algorithm is applied to select more relevant features between them according to the part of speech. Finally, we test our proposed framework on both MSVD and MSR-VTT datasets and achieve competitive performance compared with other state-of-the-art methods.

1. Introduction

Video captioning which means describing videos with natural language sentences automatically, poses an important challenge in computer vision and has plenty of applications such as video searching and event detection.

Video captioning, which aims at building the mapping from visual contents to texts, involves understanding both visual concepts and temporal dynamics. Previous work [1] has shown the potential of vision and text joint modeling for image classification. However, unlike still images, a video which comprises of a sequence of images contains more information about visual concepts and the interactions between them. It is very difficult for existing methods [2–6] to capture all the information over a long period as a single visual representation. Yao et al. first attempt to focus on a subset of video frames at each time step with a soft attention mechanism, which is driven by a LSTM-based decoder [7]. The LSTM-based decoder integrates both previously generated words and visual contents to guide visual attention and caption generation. Experiments have shown the effectiveness of the attention mechanism for video captioning.

However, temporal dynamics has not been well captured by the attention mechanism. For example, the temporal order of input video frames doesn't influence the captioning results with only the attention mechanism, which doesn't meet our common sense. Preprocessing of CNN features with LSTM helps to incorporate the temporal dynamics to some extent. Unfortunately, previous work has pointed out the weakness of LSTM for modeling very long sequence [8] and current video captioning benchmark datasets generally include videos with long sequences.

Recently, memory networks [8–10], with the potential to capture long-term correlations in sequential problems, achieve great success in question answering [11] and dialog systems [12]. Among such investigations, Neural Turing Machine [8] shows a great advantage of memorizing long-range information. We are motivated by this and contend that it is very suited to capturing the temporal dynamics from 2D CNN features. The combination of visual concepts and temporal dynamics leads to a more intact visual representation of the video. Moreover, consider the generated caption "A group of people are playing soccer on the field". The words "a" and "of" do not have corresponding visual information and language correlations make the visual information unnecessary when gener-

* Corresponding author.
E-mail addresses: liweihfyz@sjtu.edu.cn, 624654029@qq.com (W. Li).

https://doi.org/10.1016/j.patrec.2017.10.012
0167-8655/

ating words like “of” and “on”. As a result, a feature selection algorithm is included to determine the attention on visual concepts, temporal dynamics and language model according to the part of speech.

Overall, the main contributions of this paper are:

- We first conduct memory networks, instead of commonly used LSTM, to memorize long-term temporal dynamics between video frames.
- We introduce a multimodal LSTM-based decoder which combines visual concepts and temporal dynamics together as a more intact visual representation of the video.
- A feature selection algorithm is further included to decide when to look at visual concepts or temporal dynamics and when to only rely on the language model to generate the next word.
- Our model obtains comparable results on MSVD and MSR-VTT datasets with other state-of-the-art methods.
- We perform an extensive qualitative analysis on the feature selection weights of the decoder to confirm the importance of the feature selection algorithm and the superiority of memory networks over capturing temporal dynamics.

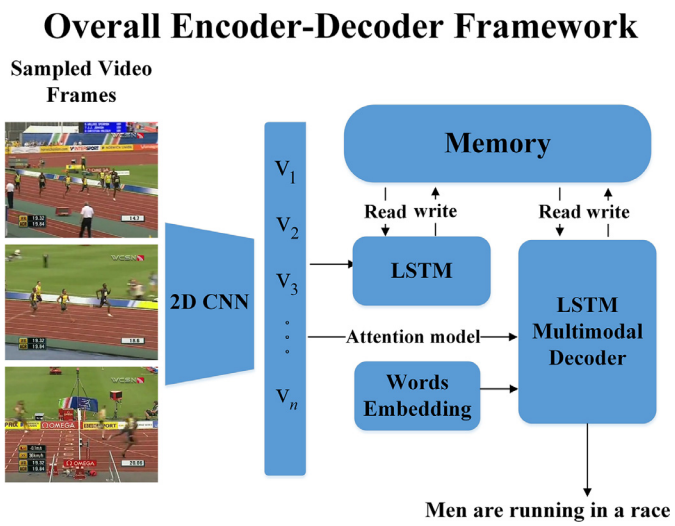

Fig. 1. Overall architecture of our proposed model. Attention weighted CNN features and temporal dynamics read from the memory matrix are fed into the LSTM-based multimodal decoder.

2. Related work

2.1. Video captioning

Video captioning draws more and more attention due to its importance in bridging vision and language. To solve this problem, variety of methods have been proposed, which can be mainly divided into two categories. The first category focuses on the identification of (subject, verb, object) triplets with visual classifiers, and captions are generated by combining predicted triplets with predefined sentence templates [13,14]. The template-based approaches obviously cannot express the richness of language and have the limited potential to unseen data. The second category is inspired by the encoder-decoder architecture originally used in machine translation [15]. Venugopalan et al. first apply it to generate descriptions of video clips with a CNN-based encoder and a LSTM-based decoder [16]. However, the temporal information of videos is inevitably left out with simple mean-pooling of CNN features. Thus, Xu et al. additionally utilize a recurrent neural network to capture the temporal dynamics [17]. Moreover, Zhu et al. train a Multirate Visual Recurrent Model (MVRM) to better incorporate both past and future temporal information [18]. Other works have then followed these kinds of approaches.

Recently, researchers begin to improve the encoder-decoder architecture by significantly changing their structures. A hierarchical framework containing a sentence generator and a paragraph generator is proposed by Yu et al. for the sentence decoder [19]. In contrast, Yao et al. first introduce a soft attention mechanism into video captioning to weight input CNN features given previously generated words [7]. Moreover, Pan et al. propose to encode videos through hierarchical recurrent neural encoders (HRNE) along with the soft attention mechanism [20]. We integrate attention weighted CNN features and temporal dynamics into a multimodal decoder for more intact visual representations in our paper. Instead of size-limited LSTM, memory networks, which have great advantages of memorizing long-term information, are used in our multimodal architecture to better capture the temporal dynamics.

2.2. Memory networks

To improve the capacity of memory cells in traditional neural networks, Graves et al. propose a Neural Turing Machine (NTM) with an external memory matrix. The memory matrix interacts with the internal states of neural networks by reading and writing which relied on a unique addressing mechanism [8]. Compared with LSTM, NTM has shown a superior potential of both storage and accessing of long-term information. Besides the memory matrix in NTM, memory is also modeled as a differentiable list, a queue or a deque [21,22]. Rather than exploring different forms of dynamic storages, Weston et al. first model the static memory to storage long-term information [10]. These memory networks have been successfully applied to long-term dependency modeling related tasks. Recently, Wang et al. introduce the memory networks into video captioning by proposing a multimodal memory network. It memorizes long-term information including both previously generated words and the visual attention history to guide the attention mechanism [23]. In this paper, we exploit the potential of Neural Turing Machine to capture the temporal dynamics from 2D CNN features.

3. The proposed approach

The overall architecture of our proposed model is shown in Fig. 1. In this section, we will first introduce our video encoder and then show the specific structure of the multimodal LSTM-based decoder. Finally, their interactions with the external memory matrix will be discussed in detail. To make the introduction more readable, we summarize important notations and their different meanings in Table 1.

3.1. Video encoder

Convolutional neural networks (CNNs) have been widely used in various computer vision tasks, e.g., image classification [2,3,24,25] and object detection [26,27]. CNN models pre-trained on large image datasets can be directly used as feature extractors for other tasks. In our model, we preprocess sampled video frames with the pre-trained CNNs. We denote the sampled CNN feature vectors by $V = \{v_1, v_2, v_3, \ldots, v_n\}$, where n is the number of the sampled frames.

RNNs are often used to extract temporal information between video frames. Although LSTM can well handle “the vanishing and exploding gradient problem” [28], it can’t memorize long-range temporal information due to the limited capacity of memory cells. Since Neural Turing Machine [8] can capture very long-term information with an external memory matrix, we utilize it as a further

Table 1
Important notations of our method.

Notation	Meaning
$V = \{v_i\}$	Sampled CNN feature vectors
z_a^t	Attention weighted CNN features
z_m^t	Contents read from memory
β_1^t	Selection weight of z_a^t
β_2^t	Selection weight of z_m^t
h_t^e, h_t^d	Hidden states of LSTM
M_t	External memory matrix
$w_t^{re}, w_t^{we}, w_t^{rd}, w_t^{wd}$	Weighting vectors generated with the addressing Mechanism
a_t^{we}, a_t^{wd}	Add vectors generated with a fully-connected layer
e_t^{we}, e_t^{wd}	Erase vectors generated with a fully-connected layer
$k_t, k_\beta^t, g_t, s_t, \gamma_t$	Parameters of addressing mechanism generated With a fully-connected layer

encoder of the CNN features to extract the temporal dynamics. The same memory matrix interacts with the decoder to provide additional temporal information.

3.2. Multimodal LSTM-based decoder

Commonly, different encoder architectures are used in different tasks, like RNN in machine translation and CNN in image captioning. However, RNN-based framework is usually applied to generate a sequence of words whether in translating or in captioning. Similar to [7,15], we utilize a LSTM-based architecture in the decoder to generate texts. LSTM units at time step t normally consist of a single memory cell, an input activation function, an output activation function and three gates (input, forget and output). The memory cell c_t records the history of all observed inputs, by recurrently summarizing the previous cell state c_{t-1} and cell input g_t by forget gate f_t and input gate i_t, respectively. The output gate o_t modulates the state of memory cell c_t to output the hidden state h_t. Different with commonly used unimodal LSTM, we incorporate attention weighted CNN features z_a^t and temporal information z_m^t read from the memory matrix as additional inputs to our multimodal LSTM of the decoder for combining visual concepts with temporal dynamics.

Visual concepts are usually localized in a subset of video frames. With the CNN feature vector $V = \{v_1, v_2, v_3, \ldots, v_n\}$ extracted from the sampled frames, instead of feeding them directly to the decoder by simple average pooling [16], we apply a soft attention mechanism to select more relevant visual features at each time step t. Specially, the soft attention mechanism first computes the unnormalized relevance score of i-th CNN feature v_i given all generated words $y_{<t}$. We utilize the LSTM hidden state of decoder h_{t-1}^d to guide attention as in [7]. Thus, the unnormalized relevance score $e_i^{(t)}$ can be described as follows:

$$e_i^t = w^T \tanh(W_a h_{t-1}^d + U_a v_i + b_a), \tag{1}$$

Where w, W_a, U_a, b_a are the parameters that are learned together with all the other parameters in the training network. After computing the unnormalized relevance score $e_i^{(t)}$, we use the softmax activation to get the attention weight $\alpha_i^{(t)}$,

$$\alpha_i^t = \frac{\exp(e_i^t)}{\sum_{j=1}^n \exp(e_j^t)}. \tag{2}$$

Finally, we weight the CNN feature vector V with the attention weights to generate the final weighted CNN features z_a^t,

$$z_a^t = \sum_{i=1}^n \alpha_i^t v_i. \tag{3}$$

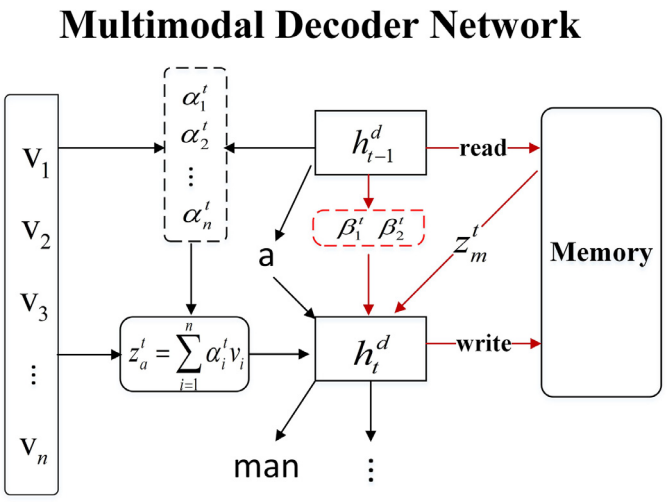

Fig. 2. Detailed architecture of multimodal decoder. Feature selection weights β_i^t are emitted by hidden state of decoder $\mathbf{h}_t^d$ to weight attention based CNN features and temporal dynamics read from memory.

In order to better incorporate the temporal dynamics when generating texts, the decoder further reads the temporal information z_m^t from the memory matrix as another input. We will introduce the detailed interactions with the memory matrix in the next section. However, as mentioned in [29], when the high-level visual representations are irrelevant to the generating word, e.g., "of", the visual representations can act as noise for the decoder. Thus, the feature selection weights β_i^t are additionally emitted by the LSTM hidden state of decoder h_t^d to choose more relevant features between them. We utilize the sigmoid nonlinearly instead of softmax function in the soft attention mechanism to make β_i^t in the range (0, 1),

$$\beta_t^i = \delta(W_\beta^i h_{t-1}^d + b_\beta^i), i \in \{1, 2\}. \tag{4}$$

The detailed architecture of our decoder is shown in Fig. 2. With the feature selection weights β_i^t, we integrate the soft attention weighted CNN features z_a^t and the contents read from memory z_m^t together into the multimodal decoder to combine visual concepts with temporal dynamics,

$$i_t^d = \delta(W_i^d[E(y_{t-1}), h_{t-1}^d, \beta_1^t z_a^t, \beta_2^t z_m^t] + b_i^d), \tag{5}$$

$$f_t^d = \delta(W_f^d[E(y_{t-1}), h_{t-1}^d, \beta_1^t z_a^t, \beta_2^t z_m^t] + b_f^d), \tag{6}$$

$$o_t^d = \delta(W_o^d[E(y_{t-1}), h_{t-1}^d, \beta_1^t z_a^t, \beta_2^t z_m^t] + b_o^d), \tag{7}$$

$$g_t^d = \delta(W_g^d[E(y_{t-1}), h_{t-1}^d, \beta_1^t z_a^t, \beta_2^t z_m^t] + b_g^d), \tag{8}$$

$$c_t^d = f_t^d \odot c_{t-1}^d + i_t^d \odot g_t^d, \tag{9}$$

$$h_t^d = o_t^d \odot \phi(c_t^d), \tag{10}$$

where $E(y_{t-1})$ represents the embedding vector of previous word y_{t-1}, $[\ldots]$ denotes vector concatenation, $\delta(x) = \frac{1}{1+exp^{-x}}$ is the sigmoid function, $\phi(x) = \frac{exp^x - exp^{-x}}{exp^x + exp^{-x}}$ is the hyperbolic tangent function and $\odot$ means matrix multiplication. All W, b are learnable parameters.

For clear illustration, the process of the decoder mentioned above can be abbreviated as follows:

$$h_t^d = \varphi([E(y_{t-1}), h_{t-1}^d, \beta_1^t z_a^t, \beta_2^t z_m^t]), \tag{11}$$

where [...] denotes vector concatenation and φ represents the updates of LSTM units.

Similar to [30], we combine the previous word embedding $E(y_{t-1})$, two different visual features z_a^t and z_m^t with the output hidden state h_t^d to generate the probability distribution over the word space,

$$p_t = \text{softmax}(W_p \tanh(E(y_{t-1}) + U_p[h_t^d, z_a^t, z_m^t])), \tag{12}$$

where [...] denotes the vector concatenation and U_p, W_p are all learnable parameters. We omit the bias terms to make the equation more readable.

In general, the model is trained to maximize the log-likelihood over the whole training set,

$$\max_{\theta} \sum_{t=1}^{T} \log p_\theta(y_t | z_a^t, z_m^t, y_{<t}), \tag{13}$$

where θ represents the model parameters, y_t means the t-th word, z_a^t and z_m^t are the two visual features at time step t and $y_{<t}$ means all generated words before step t.

As for sentence generation, two approaches are commonly used. The first approach is to sample next word based on the probability distribution over the word space until the end sign is sampled. The second approach is to select top-k best sentences at each time step and set them as the candidates for the next step to generate next top-k sentences. To generate more accurate and concise descriptions, we adopt the second approach with $k = 5$.

3.3. Interactions with the memory matrix

Although LSTM can well handle "the vanishing and exploding gradient problem", it can't deal with the long-period dependency between video frames. We utilize Neural Turing Machine [8] to memorize the long-term temporal information. Thus, the encoder and the decoder shares an external memory matrix which supplies additional temporal dynamics while generating texts.

We suppose the size of the external memory matrix M_t is $\mathrm{N} \times \mathrm{M}$ at time step t, where N is the memory location size and M is the memory vector size at each location. The memory matrix interacts with the encoder and the decoder through reading and writing. The encoder first reads memory contents r_t^{re}, together with input CNN features v_t, to update the hidden state of encoder h_t^e. Then the hidden state h_t^e summarizing visual information is written back to the memory matrix. After interactions with the encoder, the temporal dynamics has been written to the external memory matrix. Then, the decoder reads out the temporal information z_m^t from the memory matrix to generate texts. The text correlations of long sentence may be left out with size-limited LSTM, so we write the hidden state of decoder h_t^d back to the memory matrix for keeping long-range semantic consistency.

Extracting temporal dynamics to update memory in the encoder Before generating texts with the LSTM-based decoder, our encoder extracts the temporal dynamics between sampled CNN features $V = \{v_1, v_2, v_3, \ldots, v_n\}$ to update the shared memory matrix. At step t ($t = 1, 2, 3, \ldots, n$), we first read contents r_t^{re} from the memory matrix to update the hidden state of encoder h_t^e. Supposing we denote the read weighting vector of encoder by w_t^{re}, which is emitted by the previous hidden state h_{t-1}^e, then the length M read vector r_t^{re} is the weighting sum of previous memory vectors $M_{t-1}(i)$,

$$r_t^{re} = \sum_{i=1}^{N} w_t^{re}(i) M_{t-1}(i). \tag{14}$$

The updates of the hidden state of encoder h_t^e can then be simply represented similar to Eq. (11):

$$h_t^e = \varphi([h_{t-1}^e, v_t, r_t^{re}]), \tag{15}$$

where [...] denotes vector concatenation, φ represents the updates of LSTM units and r_t^{re} is the content read from memory.

After summarizing the visual representations with the hidden state h_t^e, it is further written back to the memory for long-term memorization of the temporal dynamics. We denote the write weighting vector of encoder, erase vector of encoder and add vector of encoder by w_t^{we}, e_t^{we} and a_t^{we}, respectively. All of them are emitted by the hidden state h_t^e. The elements of the erase vector e_t^{we} all lie in the range (0, 1). The updates of the memory vector $M_t(i)$ can then be represented as follows:

$$M_t(i) = M_{t-1}(i)[1 - w_t^{we}(i) e_t^{we}] + w_t^{we}(i) a_t^{we}. \tag{16}$$

Reading contents from memory to generate texts in the decoder After writing the temporal dynamics to the shared memory matrix, the contents are read out as additional information to generate texts in the LSTM-based decoder. Assuming the read weighting vector of decoder at time step t is w_t^{rd}, which is emitted by the hidden state h_{t-1}^d, the temporal information z_m^t read from the memory matrix can then be computed as a weighting sum of memory vectors $M_t(i)$,

$$z_m^t = \sum_{i=1}^{N} w_t^{rd}(i) M_t(i), \tag{17}$$

where $t = 1, 2, 3, \ldots, T$ for the decoder and T is the word count of the caption. Then with the soft attention weighted visual features z_a^t, the hidden state of decoder h_t^d is updated as Eq. (11):

$$h_t^d = \varphi([E(y_{t-1}), h_{t-1}^d, \beta_1^t z_a^t, \beta_2^t z_m^t]), \tag{18}$$

where [...] denotes vector concatenation, φ represents the updates of LSTM units and β_i^t means the feature selection weights.

Additionally, semantic correlations between words of a long sentence may not be well memorized by LSTM units. Thus, we write the hidden state h_t^d back to the memory matrix to memorize long-range semantic information. We denote the write weighting vector of decoder, erase vector of decoder and add vector of decoder by w_t^{wd}, e_t^{wd} and a_t^{wd}, respectively. All of them are emitted by the hidden state of decoder h_t^d. Then the memory vector $M_t(i)$ at every location can be updated as follows:

$$M_t(i) = M_{t-1}(i)[1 - w_t^{wd}(i) e_t^{wd}] + w_t^{wd}(i) a_t^{wd}. \tag{19}$$

Addressing mechanism Reading and writing both need weighting vectors w_t, which means $w_t^{re}, w_t^{we}, w_t^{rd}$ and w_t^{wd} in our model, to determine the attention on each memory location. Thus, we use the addressing mechanism of [8] to generate the weighting vectors based on the contents of the external memory matrix M_t and the LSTM hidden state h_t. It consists of the content-based addressing and the location-based addressing. For the content-based addressing, a normalized weighting $w_t^c(i)$ is generated through a similarity measure $K[.,.]$ between a length M key vector k_t and each memory vector $M_t(i)$,

$$w_t^c(i) = \frac{\exp(k_\beta^t K[k_t, M_t(i)])}{\sum_{j=1}^{N} \exp(k_\beta^t K[k_t, M_t(j)])}, \tag{20}$$

where k_β^t is the key strength used to amplify or attenuate the precision of focus.

Besides the content-based addressing, the location-based addressing is designed to facilitate both simple iterations and random-access jumps across rows of the memory matrix. It consists of interpolation, convolutional shift and sharpening. First, a scalar interpolation gate g_t in the range (0, 1) is introduced to combine the weighting vector of previous step w_{t-1},

$$w_t^g = g_t w_t^c + (1 - g_t) w_{t-1}. \tag{21}$$

Then, given a shift weighting s_t with a length less than N, the convolutional shift is implemented through following circular convolution:

$$\tilde{w}_t(i) = \sum_{j=0}^{N-1} w_t^g(j) s_t(i-j). \tag{22}$$

In order to attenuate the blurring impact of convolutional shift, $\gamma_t \geq 0$ is further introduced to sharpen the weighting vector as follows:

$$w_t(i) = \frac{\tilde{w}_t(i)^{\gamma_t}}{\sum_j \tilde{w}_t(j)^{\gamma_t}}. \tag{23}$$

Finally, similar to add vector a_t and erase vector e_t, addressing parameters, k_t, k_β^t, g_t, s_t and γ_t, are all emitted by the hidden state h_t with fully-connected layers.

4. Experiments and preprocessing

4.1. Datasets

We evaluate our model on two video description datasets: Microsoft Research Video Description Corpus (MSVD) [31] and Microsoft Research-Video to Text (MSR-VTT) [32].

4.1.1. Microsoft Research Video Description Corpus (MSVD)

The MSVD dataset has 1970 open domain clips collected from YouTube and annotated using a crowd sourcing platform. It contains multiple topics including sports, cooking and movies. Each video has multiple natural language descriptions and the dataset has more than 80,000 captions in total. We follow the splits made by [7], separating the datasets into a training set of 1200 clips, a validation set of 100 clips and a test set of 670 clips.

4.1.2. Microsoft research-Video to text (MSR-VTT)

Microsoft Research-Video to Text (MSR-VTT) is a large-scale dataset to public for bridging video and language. The dataset contains 10,000 video clips and 200,000 sentences. Each video is labeled with 20 sentences. Following the split of [32], we divide the whole dataset into a training set of 6513 clips, a validation set of 497 clips and a test set of 2990 clips.

4.2. Evaluation metrics

We report the performance of our proposed model on three standard evaluation metrics: BLEU [33], METEOR [34] and CIDER [35]. Specially, BLEU computes the N-gram similarity, and METEOR focuses on the mapping between unigrams while CIDER emphasizes more on the human judgment of similarity between generated sentences and references. Based on the evaluation of [35], METEOR is always better than BLEU in terms of consistency with human judgment and CIDER gets similar results compared with METEOR. Thus, similar to [20], we use the evaluation server introduced in [36] and select the best model based on METEOR evaluation metric.

4.3. Preprocessing

The datasets have videos of various frame lengths. Thus, we subsample the video frames to ensure a fixed length of K. For those videos with less than K frames, we append the feature vectors with blank vectors. Following the setting of [23], we set K to 28 for MSVD and to 40 for MSR-VTT for a fair comparison. We extract features with pretrained 2D CNN networks, e.g., GoogleNet [2] for MSVD, VGG-19 [3] and ResNet-152 [24] for MSR-VTT. Moreover, we convert all captions to lower case and tokenize the sentences using the PTBTokenizer in Stanford CoreNLP tools [37]. This yields a vocabulary of 13,010 in size for MSVD and 29,019 in size for MSR-VTT.

Table 2
Experiment results on MSVD .

Models	BLEU	METEOR	CIDER
Ours-google	0.480	0.316	**0.688**
SA [7]	0.419	0.296	0.517
LSTM-E [41]	0.417	0.299	-
h-RNN [19]	0.443	0.311	0.621
HRNE [20]	0.438	**0.331**	-
M^3-google [23]	**0.5117**	0.3147	-

4.4. Experimental settings

We empirically set the dimension of hidden states to 512. Additionally, the length of memory vector is 1000 along with 10 memory locations which means the size of the external memory matrix is 10×1000. All experiments are conducted with a training mini-batch size of 64 and L2-decay term of 0.0001. Dropout [38] with a rate of 0.5 is utilized in our experiments as another regularization. Moreover, we train our model with the Adadelta [39] optimization algorithm and a clipnorm of 10. All of our experiments are implemented with Keras and Theano [40].

4.5. Performance comparison

In order to verify the effectiveness of our proposed model, we test our model on both MSVD and MSR-VTT datasets for video captioning. We choose three different evaluation metrics for comparison.

4.5.1. Experiment results on MSVD

We compare our model with other five state-of-the-art methods [7,19,20,23,41] using single visual feature for video captioning. The experiment results of BLEU, METEOR and CIDER evaluation metrics are shown in Table 2. Here we obtain comparable results with other state-of-the-art methods. Compared with SA [7], our method additionally utilizes memory networks to capture temporal dynamics for more intact visual representations. We gain a large margin performance improvement over SA, which implies the effectiveness of the memory networks for video captioning. Besides SA, M^3 [23] is the most similar method to ours. It also has an external memory matrix for the LSTM-based decoder. M^3, however, only utilize the external memory matrix for long-range visual-textual information interactions. The temporal dynamics has not been well captured since the temporal order of input video frames doesn't influence the captioning results. We first exploit the potential of memory networks to extract temporal dynamics. Our model outperforms all other methods except M^3-google [23] in terms of BLEU. It is because BLEU emphasizes more on N-gram similarity while our model combines visual concepts with temporal dynamics directly which sacrifices the semantic consistency to some extent. However, we outperform M^3 on all metrics on MSR-VTT [32] dataset with longer videos, which implies the importance of temporal dynamics for video captioning. Also, we achieve better performance in METEOR compared with other methods except HRNE [20] because HRNE focus more on building a fine-grained hierarchical structure for captioning. It is beneficial for appropriate mapping of unigrams which is crucial for the METTOR metric.

In Fig. 3, two video clips from the test set are shown. We can clearly see that the generated captions correspond well with the video clips. From the top panel, when the word "pot" is to be generated, the model mainly focuses on the third frame and the pot occupies the vast majority of that frame. Similarly, on the bottom

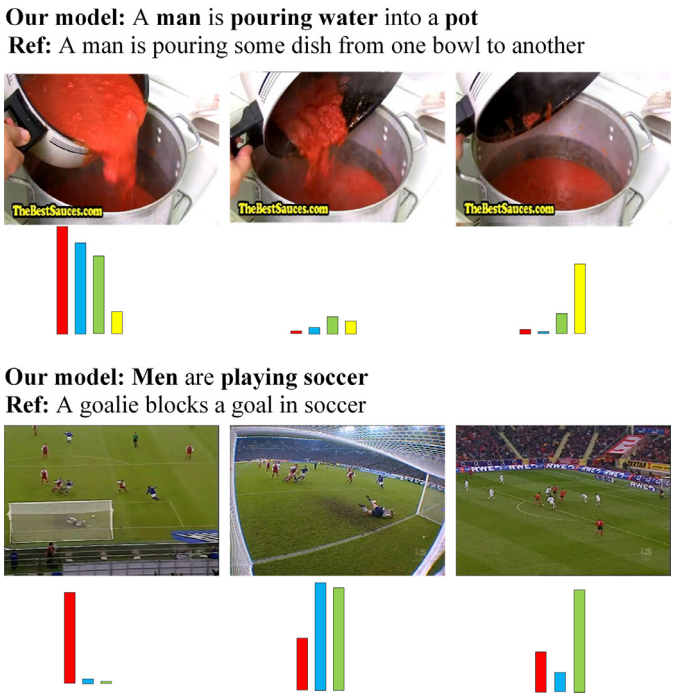

Fig. 3. Sampled videos and their corresponding generated and ground-truth descriptions. The bar plot under each frame represents the attention weight α_i^t when corresponding boldfaced word is to be generated.

Table 3
Experiment results on MSR-VTT .

Models	B@1	B@2	B@3	B@4	METEOR
Ours-V	0.752	0.606	0.472	0.358	0.256
Ours-R	**0.761**	**0.621**	**0.491**	**0.375**	**0.264**
SA-V [7]	0.678	0.554	0.429	0.347	0.231
M^3-V [23]	0.702	0.566	0.448	0.350	0.246

panel, when the word "soccer" is about to be generated, the model pays more attention on the second frame and the third frame where the goal and the stand are highly visible showing it's a soccer game. The detailed visualization of attention weights shows the effectiveness of the soft attention mechanism for visual concepts extraction.

4.5.2. Experiment results on MSR-VTT

MSR-VTT is a recently released benchmark dataset and there are few methods tested on this dataset. We compare our model with two most similar methods, SA [7] and M^3 [23] using single visual feature. The results are shown in Table 3. Ours-V, SA-V and M^3-V all use VGG-19 feature and our model outperforms other methods on all metrics. Also, we report even better result with Resnet-152 feature as Ours-R. Temporal dynamics is not well captured by SA and M^3 since the temporal order of input video frames doesn't influence the captioning results of them. The improved performance of our model confirms the importance of temporal dynamics for captioning of long videos.

4.6. Qualitative analysis of feature selection weights

In order to illustrate the importance of the feature selection algorithm and the superiority of memory networks over capturing temporal dynamics, we sample two video clips and their corresponding generated captions to visualize the feature selection weights when generating different words. From the upper half of Fig. 4, we show that when the word "walking" is about to

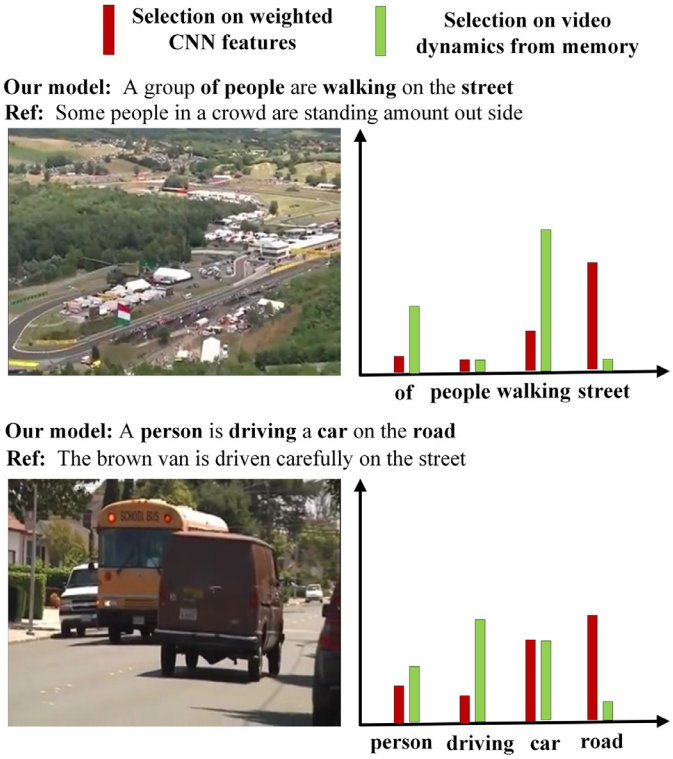

Fig. 4. Additional feature selection weights of sampled videos. The red box β_1^t means selection on weighted CNN features and the green box β_2^t indicates selection on temporal dynamics from memory. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

be generated, the model focuses more on the temporal dynamics read from the memory matrix. Similarly, it pays more attention on the weighted CNN features to generate words "street" and "road". Moreover, when visual irrelevant word "of" is to be generated, both selection weights are of relative small value so the model can only focus on the language model. However, the correlation between selection weights and characteristic of word is sometimes not like that. For example, when the word "people" is to be generated, β_1^t and β_2^t are both of small value. It's because the connection between "people" and previously generated words is very strong and "people" can be easily inferred with "a group of".

With the feature selection algorithm, the model can select more related features based on the part of speech. Also, when generating visual irrelevant words, the model tends to only rely on the language model. With the visualization of the feature selection weights, we show the details of the feature selection algorithm and verify the effectiveness of our proposed model.

5. Conclusion and future work

In this paper, we demonstrate the combination of visual concepts and temporal dynamics is crucial for video captioning. Our model preserves an intact video structure, resulting in a more accurate description for the video. First, we incorporate an attention mechanism and memory networks into a multimodal decoder for capturing more integrated contents of the video. On the one hand, the attention mechanism attends to select a subset of relevant frames based on previously generated words. It is very important for visual concepts extraction. On the other hand, the memory networks, which have a significant advantage of memorizing long-term information, provides additional temporal dynamics for our multimodal decoder. Through the visualization of generated descriptions, we verify the effectiveness of the combination of these

two methods. Second, a feature selection algorithm is additionally applied to select more relevant features given previously generated words. For example, when generating visual irrelevant words like “of”, the model should only focus on the language model rather than the visual contents according to the common sense. The feature selection algorithm further boosts our model performance. Finally, we report our performance on two video description datasets, MSVD and MSR-VTT and obtain comparable results with other state-of-the-art methods.

According to Pan et al. [20], GRU [42] could be a better option for temporal information modeling compared with LSTM. We will explore the effectiveness of GRU or LSTM variants for our multimodal decoder in our future work. Last but not least, Zhu et al. recently proposed an effective joint language-vision modeling approach [43]. We will include this new method into our architecture to better model temporal information.

References

[1] Y. Yan, F. Nie, W. Li, C. Gao, Y. Yang, D. Xu, Image classification by cross-media active learning with privileged information, TMM 18 (12) (2016) 2494–2502.
[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, CVPR, 2015.
[3] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, ICLR, 2015.
[4] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, NIPS, 2012.
[5] S. Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, TPAMI 35 (1) (2013) 221–231.
[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, ICCV, 2015.
[7] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, A. Courville, Describing videos by exploiting temporal structure, CVPR, 2015.
[8] A. Graves, G. Wayne, I. Danihelka, Neural turing machines, arXiv preprint arXiv:1410.5401 (2014).
[9] S. Sukhbaatar, J. Weston, R. Fergus, et al., End-to-end memory networks, NIPS, 2015.
[10] J. Weston, S. Chopra, A. Bordes, Memory networks, arXiv preprint arXiv:1410.3916 (2014).
[11] C. Xiong, S. Merity, R. Socher, Dynamic memory networks for visual and textual question answering, ICML, 2016.
[12] J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. Miller, A. Szlam, J. Weston, Evaluating prerequisite qualities for learning end-to-end dialog systems, ICLR, 2016.
[13] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, K. Saenko, Youtube2text: recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition, CVPR, 2013.
[14] N. Krishnamoorthy, G. Malkarnenkar, R.J. Mooney, K. Saenko, S. Guadarrama, Generating natural-language video descriptions using text-mined knowledge., AAAI, 2013.
[15] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, NIPS, 2014.
[16] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, K. Saenko, Translating videos to natural language using deep recurrent neural networks, NAACL HLT, 2015.
[17] H. Xu, S. Venugopalan, V. Ramanishka, M. Rohrbach, K. Saenko, A multi-scale multiple instance video description network, ICCV, 2015.
[18] L. Zhu, Z. Xu, Y. Yang, Bidirectional multirate reconstruction for temporal modeling in videos, CVPR, 2017.
[19] H. Yu, J. Wang, Z. Huang, Y. Yang, W. Xu, Video paragraph captioning using hierarchical recurrent neural networks, CVPR, 2016.
[20] P. Pan, Z. Xu, Y. Yang, F. Wu, Y. Zhuang, Hierarchical recurrent neural encoder for video representation with application to captioning, CVPR, 2016.
[21] A. Joulin, T. Mikolov, Inferring algorithmic patterns with stack-augmented recurrent nets, NIPS, 2015.
[22] E. Grefenstette, K.M. Hermann, M. Suleyman, P. Blunsom, Learning to transduce with unbounded memory, NIPS, 2015.
[23] J. Wang, W. Wang, Y. Huang, L. Wang, T. Tan, Multimodal memory modelling for video captioning, arXiv preprint arXiv:1611.05592 (2016).
[24] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR, 2016.
[25] X. song Tang, K. Hao, H. Wei, Y. Ding, Using line segments to train multi-stream stacked autoencoders for image classification, PR Lett. 94 (2017) 55–61.
[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, ECCV, 2016.
[27] M. Kmie, A. Glowacz, Object detection in security applications using dominant edge directions, PR Lett. 52 (2015) 72–79.
[28] J. Kolen, S. Kremer, Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies, Wiley-IEEE Press, 2001, pp. 237–243. https://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5264952&newsearch=true&queryText=Gradient%20Flow%20in%20Recurrent%20Nets:%20The%20Difficulty%20of%20Learning%20LongTerm%20Dependencies.
[29] J. Lu, C. Xiong, D. Parikh, R. Socher, Knowing when to look: Adaptive attention via A visual sentinel for image captioning, CVPR, 2017 https://openaccess.thecvf.com/CVPR2017.py.
[30] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R.S. Zemel, Y. Bengio, Show, attend and tell: neural image caption generation with visual attention, ICML, 2015.
[31] D.L. Chen, W.B. Dolan, Collecting highly parallel data for paraphrase evaluation, ACL, 2011.
[32] J. Xu, T. Mei, T. Yao, Y. Rui, Msr-vtt: A large video description dataset for bridging video and language, CVPR, 2016.
[33] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, ACL, 2002.
[34] M. Denkowski, A. Lavie, Meteor universal: Language specific translation evaluation for any target language, EACL, 2014.
[35] R. Vedantam, C. Lawrence Zitnick, D. Parikh, Cider: Consensus-based image description evaluation, CVPR, 2015.
[36] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, C.L. Zitnick, Microsoft coco captions: data collection and evaluation server, arXiv preprint arXiv:1504.00325(2015).
[37] C.D. Manning, M. Surdeanu, J. Bauer, J.R. Finkel, S. Bethard, D. McClosky, The stanford corenlp natural language processing toolkit., ACL, 2014.
[38] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting., JMLR 15 (1) (2014) 1929–1958.
[39] M.D. Zeiler, Adadelta: an adaptive learning rate method, arXiv preprint arXiv:1212.5701(2012).
[40] T.T.D. Team, R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, et al., Theano: a python framework for fast computation of mathematical expressions, arXiv preprint arXiv:1605.02688(2016).
[41] Y. Pan, T. Mei, T. Yao, H. Li, Y. Rui, Jointly modeling embedding and translation to bridge video and language, CVPR, 2016.
[42] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, EMNLP (2015).
[43] L. Zhu, Z. Xu, Y. Yang, A.G. Hauptmann, Uncovering the temporal context for video question answering, IJCV (2017), doi:10.1007/s11263-017-1033-7.