MAM-RNN: Multi-level Attention Model Based RNN for Video Captioning

Xuelong Li[1], **Bin Zhao**[2], **Xiaoqiang Lu**[1]
[1]Xian Institute of Optics and Precision Mechanics, Chinese Academy of Sciences,
Xian 710119, P. R. China
[2]School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL),
Northwestern Polytechnical University, Xian 710072, P. R. China
xuelong_li@opt.ac.cn, binzhao111@gmail.com, luxiaoqiang@opt.ac.cn

Abstract

Visual information is quite important for the task of video captioning. However, in the video, there are a lot of uncorrelated content, which may cause interference to generate a correct caption. Based on this point, we attempt to exploit the visual features which are most correlated to the caption. In this paper, a Multi-level Attention Model based Recurrent Neural Network (MAM-RNN) is proposed, where MAM is utilized to encode the visual feature and RNN works as the decoder to generate the video caption. During generation, the proposed approach is able to adaptively attend to the salient regions in the frame and the frames correlated to the caption. Practically, the experimental results on two benchmark datasets, i.e., MSVD and Charades, have shown the excellent performance of the proposed approach.

1 Introduction

The task of video captioning is to automatically generate a sentence to describe the video content [Venugopalan *et al.*, 2015a]. Recently, this task has drawn increasing attention, because it enables many important applications, including video title generation, human-robot interaction, content-based video retrieval and so on.

Essentially, video captioning is a sequence to sequence task, which transfers the data from frame sequence to word sequence. Recently, benefiting from the development of deep learning, especially *Recurrent Neural Network* (RNN), video captioning has achieved inspiring results [Zeng *et al.*, 2016; Shetty and Laaksonen, 2015]. Practically, most of existing RNN based approaches follow the encoder-decoder diagram. Firstly, the video features are encoded into a fixed size vector which is taken as the input to the RNN. Then, the RNN is utilized as the decoder to generate the sentence.

Practically, the visual feature input to RNN is quite important for generating correct video caption. The early approach [Venugopalan *et al.*, 2015b] simply input the average-pooled frame features to RNN. Recently, researchers have realized that there are a lot of redundant and irrelevant content in the video, which may cause interference to generate the correct caption. Based on this point, attention model is employed to selectively focus on only a few of the video frames which are relevant to the target caption [Yao *et al.*, 2015; Yu *et al.*, 2016]. However, there are still some irrelevant background information, especially when the described object is small. To address this problem, a more powerful attention model is needed to automatically attend to the most salient regions in each frame. Ideally, as depicted in Fig. 1, to generate the caption of *a dog is running around*, we hope the attention model can automatically focus on not only the frames containing dogs, but also their dog regions.

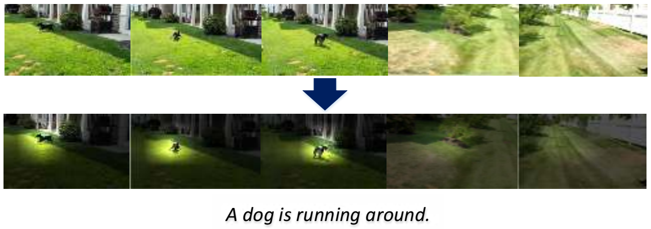


Figure 1: Our multi-level attention model can not only focus on the most correlated frames, but also attend to the most salient regions in each frame.

Following this inspiration, we design a Multi-level Attention Model (MAM) to encode the video features. It is able to adaptively focus on the most correlated visual features both in frame-level and region-level. Specifically, it has two layers. When generating the caption, the first layer learns to focus on the most salient regions in each frame, while the second layer tries to attend to the most correlated frames. Then, the video feature is encoded into a fixed size vector, and input to the RNN to generate the video caption word by word. Note that the visual feature and text feature are fed to RNN jointly, which indicates that the current word is not only determined by the current visual feature, but also the text feature of previous word.

To our knowledge, this is the first approach that jointly apply region-level and frame-level attention to the task of video captioning, which is capable of focusing on the most correlated visual features to generate the correct caption. Moreover, the results on the MSVD dataset [Guadarrama *et al.*, 2013] and Charades dataset [Sigurdsson *et al.*, 2016] have verified the effectiveness of our approach.

2 Related Works

Early works on video captioning follow a two stage pipeline [Guadarrama *et al.*, 2013; Krishnamoorthy *et al.*, 2013; Thomason *et al.*, 2014]. The first stage trains individual classifiers to identify the semantic content, i.e., objects, actions and scenes. Then, in the second stage, these semantic contents are combined with a probabilistic graphical model to generate a sentence. However, this kind of approach is insufficient to model the richness of visual and semantic information in video captioning.

Recently, benefiting from the rapid development of deep learning, video captioning has made great progress. Current approaches usually follow the CNN-RNN architecture, where *Convolutional Neural Network* (CNN) [Krizhevsky *et al.*, 2012; Simonyan and Zisserman, 2014; Szegedy *et al.*, 2015] is utilized to extract the visual feature of the video and RNN is good at modeling the sequence information of the caption. The baseline of CNN-RNN based approach is first proposed in [Venugopalan *et al.*, 2015b]. Firstly, the visual feature of the video is extracted by simply average pooling the CNN features of individual frames. Then, the visual feature is input to the *Long Short-Term Memory* (LSTM) to generate the sentence word by word. The main shortcoming of this approach is that it fails to capture the temporal information of video frames. Thus, it only works for short video clips. To exploit more powerful visual feature, [Venugopalan *et al.*, 2015a] employs a LSTM to encode the CNN feature of each frame. More recently, inspired by the success of attention model in image captioning, [Yao *et al.*, 2015; Yu *et al.*, 2016] design attention models to selectively focus on a subset of video frames, and the visual feature is generated by the weighted sum of the attended frame features. Actually, the CNN-RNN architecture extended with attention model has won the state-of-the-art performance in the task of video captioning.

3 Our Approach

As depicted in Fig. 2, our approach, MAM-RNN, can be divided into two stages. The first stage is to extract the visual feature with MAM, which is composed of the region-level attention and the frame-level attention. The second stage is to generate the video caption with RNN, i.e., LSTM in this paper [Hochreiter and Schmidhuber, 1997]. For clarity, we first provide a brief introduction about RNN, and then successively present the process of video caption generation and visual feature extraction.

3.1 Recurrent Neural Network

Recurrent Neural Network (RNN) is extended from the feedforward networks with extra feedback connections, so that it can model the sequence information. In standard RNN, given an input sequence $(x_1, x_2, \ldots, x_n)$, the output sequence $(y_1, y_2, \ldots, y_n)$ can be generated iteratively by the following equations:

$$h_t = \phi\left(W_h x_t + U_h h_{t-1} + b_h\right), \tag{1}$$

$$y_t = \phi\left(U_y h_t + b_y\right), \tag{2}$$

where h_t denotes the hidden state at time t, $\phi(\cdot)$ is the activation function, and W, U, b are parameters to be learned.

According to [Bengio *et al.*, 1994], the standard RNN is inferior at modeling long-term sequence information, meanwhile, it is hard to train due to the vanishing gradient problem. Fortunately, LSTM can make up this drawback, which is derived from the standard RNN by adding three gate layers, i.e., the input gate i_t, the forget gate f_t and the output gate o_t. In LSTM, the hidden state is calculated iteratively by

$$i_t = \sigma\left(W_{ix} x_t + U_{ih} h_{t-1} + b_i\right), \tag{3}$$

$$f_t = \sigma\left(W_{fx} x_t + U_{fh} h_{t-1} + b_f\right), \tag{4}$$

$$o_t = \sigma\left(W_{ox} x_t + U_{oh} h_{t-1} + b_o\right), \tag{5}$$

$$g_t = \phi\left(W_{gx} x_t + U_{gh} h_{t-1} + b_g\right), \tag{6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \tag{7}$$

$$h_t = o_t \odot \phi\left(c_t\right), \tag{8}$$

where all the Ws, Us and bs are training parameters.

3.2 Video Caption Generation with RNN

In this paper, the LSTM is employed as the generator of video captions. It can be observed from Equ. (3)–(8) that the hidden state h_t in LSTM is determined by the input x_t and the previous hidden state h_{t-1}. Thus, for simplicity, the calculation of hidden state is denoted as:

$$h_t = LSTM\left(x_t, h_{t-1}\right). \tag{9}$$

Specifically, the input data for the task of video captioning is combined by:

$$x_t = \left[W_V V_t, e_{t-1}\right], \tag{10}$$

where V_t represents the input visual feature at time step t. That is to say, different visual features are input to the LSTM at every time step. The details about the calculation of V_t is described in the next subsection. e_{t-1} is the text feature of the word at $t-1$. In this paper, the one-hot feature is extracted for each word, and embedded into a lower space. Besides, W_V is a training matrix which embeds the visual feature V_t and text feature e_{t-1} into the same space.

Once the hidden state h_t is computed, the probability distribution over the vocabulary is calculated as:

$$p_t = softmax\left(\tan\left(W_p\left[V_t, e_{t-1}, h_t\right] + b_p\right)\right), \tag{11}$$

where W_p and b_p are the parameters to be learned. p_t denotes the probability of each element in the vocabulary to be selected as the t-th word of the caption, which is jointly determined by the current visual feature V_t, previous text feature e_{t-1} and all the history information encoded in h_t. Practically, the dimensionality of p_t is equal to the size of the vocabulary.

Specifically, in the training procedure, the log-likelihood function is employed,

$$\Theta = \arg\max_{\Theta} \sum_{t=1}^{T} \log \Pr\left(g_t | g_{t-1}, V_t; \Theta\right), \tag{12}$$

where g_t denotes the t-th word of the reference caption, Θ stands for all the training parameters in our approach. T is the total time steps of the LSTM. Practically, if the reference

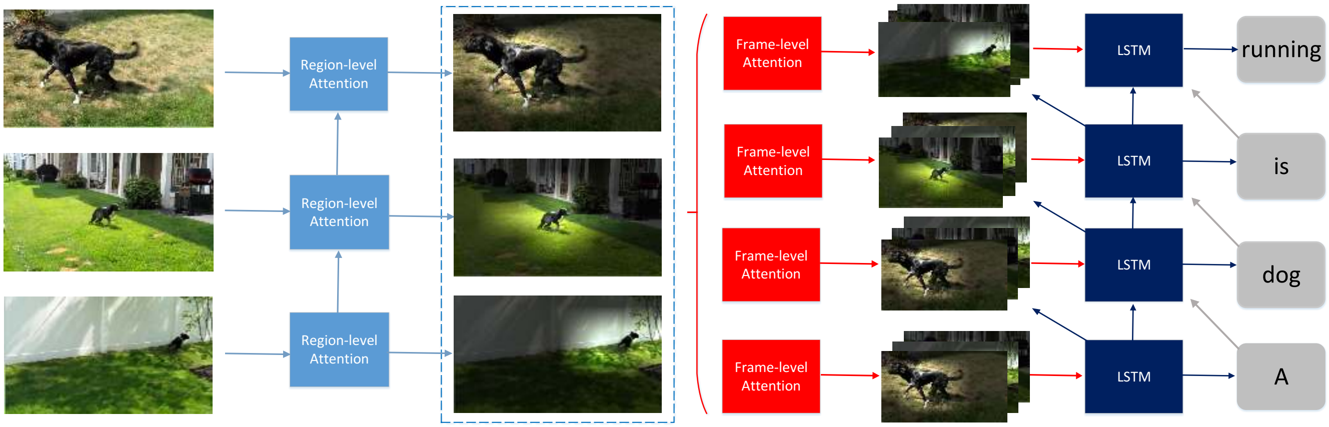


Figure 2: The architecture of our MAM-RNN. Specifically, MAM contains the region-level attention and frame-level attention, which automatically selects the most correlated visual feature as the input to RNN. The RNN, i.e., LSTM in this paper, is employed to generate the caption word by word.

caption is shorter than T, it is padded by zeros. Equ. (12) illustrates that the parameters are learned by maximizing the probability of the reference caption.

In the testing procedure, the video caption is generated word by word with the maximum value in the vector p_t, depicted in Equ. (11).

3.3 Visual Feature Extraction with MAM

The target of MAM is to generate the visual feature V_t at every time step. As aforementioned, our MAM have two layers, i.e., region-level attention layer and frame-level attention layer. In the following, we introduce the two layers detailedly.

Region-level Attention Layer

Actually, this layer is employed to generate the frame feature by emphasizing the salient regions in each frame.

Firstly, CNN is employed to extract the region features for each frame. Thus, for a certain frame i, we get a set of feature vector, $\{r_{i1}, r_{i2}, \dots, r_{im}\}$, where m denotes the number of regions.

Then, the frame feature is generated with the weighted sum of the region features,

$$f_i = \sum_{j=1}^{m} \alpha_{ij} r_{ij}, \tag{13}$$

where α_{ij} denotes the attention weight of the j-th region of frame i, which is computed by the following equations:

$$m_{ij} = w^r \tanh\left(W^r r_{ij} + U^r \alpha_{i-1,j} + b^r\right), \tag{14}$$

$$\alpha_{ij} = \exp\{m_{ij}\} / \sum_{j=1}^{k} \exp\{m_{ij}\}. \tag{15}$$

When the parameters W^r, U^r and b^r are learned, the salient regions in the frame are emphasized by α, which can reduce the influence of irrelevant and meaningless region features. It can be observed from Equ. (14) that the attention weights of regions in frame i are not only determined by their features, but also by the attention weights of the previous frame. The intuition lying behind this is that consecutive frames are quite similar with each other, so we hope the attention weight vary smoothly according to time.

Frame-level Attention Layer

In this layer, the final visual feature of the video is obtained by adaptively focusing on a subset of frames which is correlated to the video caption.

Specifically, the visual feature input to the RNN at time t, i.e., V_t, is computed by the weighted sum of the frame features,

$$V_t = \sum_{i=1}^{n} \beta_i^t f_i, \tag{16}$$

where n denotes the number of frames, β_i^t is the attention weight of frame i at time t. Ideally, if frame i is very correlated to the caption, its feature is much emphasized when generating the visual feature. Note that the β_i^ts are computed at each time step, so it is a dynamic attention layer.

Practically, to capture the sequential information, β_i^t should be determined not only by the frame feature f_i, but also by all the information before time t, including the visual feature and the text feature. Fortunately, they are encoded in the previous hidden state h_{t-1} of LSTM. Actually, according to [Yao *et al.*, 2015], the attention weight β_i^t reflects the relevance of f_i given h_{t-1}. Specifically, in this paper, the relevance score is calculated by

$$l_i^t = w^f \tanh\left(W^f f_i + U^f h_{t-1} + b^f\right), \tag{17}$$

where W^f, U^f, b^f are training parameters, and

$$h_{t-1} = LSTM\left(V_{t-1}, e_{t-2}, h_{t-2}\right). \tag{18}$$

Then, the l_i^t are normalized to

$$\beta_i^t = \exp\{l_i^t\} / \sum_{i=1}^{n} \exp\{l_i^t\}. \tag{19}$$

With frame-level attention weight β_i^t, the visual feature at time t, i.e., V_t, can automatically attend to the most correlated frame features, which can enhance the accuracy of the generated video caption.

Table 1: The results of various approaches on MSVD dataset. (The scores in bold indicate the best value.)

Metrics	CIDEr	BLEU–1	BLEU–2	BLEU–3	BLEU–4	ROUGE–L	METEOR
FGM [Thomason *et al.*, 2014]	–	–	–	–	–	–	0.239
Mean Pool [Venugopalan *et al.*, 2015b]	–	–	–	–	0.372	–	0.281
S2VT [Venugopalan *et al.*, 2015a]	0.486	0.735	0.593	0.482	0.369	0.652	0.289
SA [Yao *et al.*, 2015]	0.481	0.741	0.589	0.482	0.366	0.647	0.294
LSTM-E [Pan *et al.*, 2016]	–	0.749	0.609	0.506	0.402	–	0.295
p-RNN [Yu *et al.*, 2016]	–	0.773	0.645	0.546	**0.443**	–	0.311
MAM-RNN	**0.539**	**0.801**	**0.661**	**0.547**	0.413	**0.688**	**0.322**

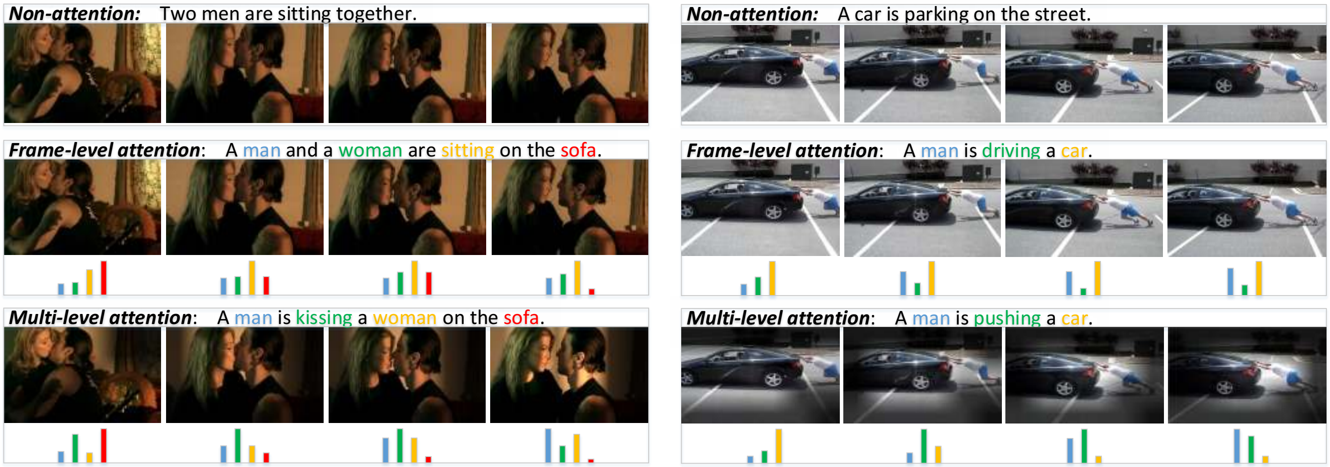

Non-attention: Two men are sitting together.
Frame-level attention: A man and a woman are sitting on the sofa.
Multi-level attention: A man is kissing a woman on the sofa.
Non-attention: A car is parking on the street.
Frame-level attention: A man is driving a car.
Multi-level attention: A man is pushing a car.

Figure 3: Example results on the MSVD dataset. The sentences above frames denote the captions generated by three variants of our approach, i.e., non-attention, frame-level attention and multi-level attention. The histograms below frames represent the frame-level attention when generating each word (distinguished by color). The brightness distribution in each frame reflects the region-level attention, i.e., the brighter regions are more emphasized.

4 Experiments

4.1 Experimental Details

Datasets

The MSVD dataset [Guadarrama *et al.*, 2013] is composed of 1970 video clips downloaded from the YouTube. Each video clip typically describes a single activity in open domain and is annotated with multi-lingual captions. In this paper, we only consider the captions in English, about 41 captions for each video and 80839 captions in total. Generally, each caption contains about 8 words.

The Charades dataset [Sigurdsson *et al.*, 2016] is more challenging, which consists of 9848 videos with an average length of 30 seconds. Different from the YouTube clips in MSVD, the videos in Charades dataset record the daily living activities in indoor scenes, like cooking, eating, using phone and so on. Actually, the scenes and activities captured in the dataset are quite diverse. Totally, the dataset provides 27847 video captions, about 3 captions for each video.

Feature Extraction

Region feature: according to previous works [Yao *et al.*, 2015; Fakoor *et al.*, 2016], the region feature is extracted from a lower convolutional layer of the CNN. Actually, to analyze the influence of different features, three popular CNNs are employed, i.e., VggNet 16 (pre-trained on ImageNet), GoogLeNet (pre-trained on ImageNet) and C3D (pre-trained on Sports-1M), where the pool5 layer (7*7*512), inception 5b layer (7*7*1024) and conv5b layer (7*7*1024) are utilized to extract the region features, respectively. It indicates that each frame is divided equally into 7*7 grid regions, and each region is represented by 512, 1024, 1024 dimensionality feature vector in the three CNNs. These region features are widely used in video captioning. Besides, for the efficiency of our approach, in each video, we just consider 160 frames generated by uniform sampling. For videos with fewer than 160 frames, we pad them with zeros.

Text feature: the video captions are preprocessed by converting all words to lower case, removing rare words and tokenizing the sentences. After preprocessing, the size of the total vocabulary is 2743 for the MSVD dataset and 1525 for the Charades dataset, respectively. Then, the one-hot feature (1-of-N coding, N denotes the number of words in the vocabulary) is extracted for each word and embedded into a 300-dimensional GloVe vector [Pennington *et al.*, 2014], which has shown great performance in word analogy task.

Evaluation Metrics

The task of video captioning share similar evaluation metrics with machine translation, such as BLEU [Papineni *et al.*, 2002], ROUGE–L [Lin and Och, 2004], CIDEr [Vedantam *et al.*, 2015], METEOR [Denkowski and Lavie, 2014], they are widely used to evaluate the quality of video captions. To provide a comprehensive evaluation, all of the above four metrics are employed in this paper, where BLEU has four versions, i.e., BLEU 1–4. Practically, the metrics are calculated based on the alignment between the predicted sentence and the reference sentence, including the word matching and semantic similarity. Generally, the higher scores indicates the higher quality of the generated caption.

4.2 Results and Discussion

Results on the MSVD Dataset

The MSVD dataset is split into a training set of 1200 videos, a validation set of 100 videos, and a testing set of the remaining 670 videos. Table 1 shows the performance of various approaches on the MSVD dataset. Note that, to provide a fair judgement, we get rid of the influence of different features. Practically, all the approaches listed in Table 1 are constrained to extract video feature with the same CNN, i.e., VggNet 16, which is the most widely used CNN in video captioning.

In Table 1, all the compared approaches follow the CNN-RNN architecture, expect for FGM. Specifically, FGM uses a factor graph to estimate the most like words based on visual detections, which gets the state-of-the-art results in non-RNN approaches. However, from the METEOR metric, it can be observed that the CNN-RNN based approaches achieve much better performance. In fact, the main difference between the compared approaches and our approach lies in the visual feature extraction. Detailedly, Mean Pool generates the visual feature by simply average pooling all the frame features, S2VT encodes the visual feature by an LSTM and just input the visual feature at the first time step. The proposed approach, MAM-RNN, achieves better results than them, because we design attention model to emphasize the most correlated features in the video, and input the visual feature at every time step. Besides, our approach also outperforms SA and p-RNN, where SA is equipped with a frame-level attention model and p-RNN employs a region-level attention model (just focus on some frame regions), respectively. It is because that our multi-level attention model combines the advantages of the two models. Therefore, it can not only focus on the most correlated frame features, but also the salient region features. Additionally, our approaches also gets better results than LSTM-E, which incorporates a visual-semantic embedding space into the LSTM.

To analyze the influence of different features to the performance of our approach, the results with different features are depicted in Table 2, where only the METEOR metric is provided. In Table 2, three popular CNNs are considered, i.e., VggNet 16, GoogLeNet, C3D. On one hand, it can be observed that all the three versions of our approach perform better than the compared approaches, even better than the approaches with combined features, e.g., S2VT combines the VggNet 16 feature of RGB frame and optical flow frame, and SA combines the frame feature of GoogLeNet and C3D. On the other hand, with C3D and GoogLeNet, our approach get better performance than VggNet 16. It is because that C3D is a 3D convolutional neural network, it can exploit the dynamic information in the video. While, GoogLeNet is deeper than VggNet 16, which indicates more powerful capability to exploit the video information. Based on the difference among the results of the three version approaches, it can be imagined that our approach can get even better performance with more powerful feature extractor.

Table 2: The results with different features on MSVD dataset.

Metrics	METEOR
Mean pool (GoogLeNet)	0.287
S2VT(RGB+FLOW VggNet 16)	0.297
SA(GoogLeNet+C3D)	0.296
LSTM-E(C3D)	0.299
p-RNN (C3D)	0.303
MAM-RNN (VggNet)	0.322
MAM-RNN (C3D)	0.325
MAM-RNN (GoogLeNet)	**0.329**

To verify the effectiveness of our Multi-level Attention Model (MAM), in Figure 3, we show a few examples about the results with different attention models, i.e., non-attention model (without attention), frame-level attention model and multi-level attention model. It can be observed that compared to the other two versions, our multi-level attention model shows obvious advantages in exploiting the accurate visual feature, where the region-level attention layer are able to automatically focus on the most salient regions in the frame, and the frame-level attention layer can automatically attend to the most correlated frames. When generating the caption, it can significantly reduce the interference caused by the irrelevant and meaningless visual feature. That is why our our MAM is effective in improving the accuracy of the video caption, reflected in the generated captions in Figure 3.

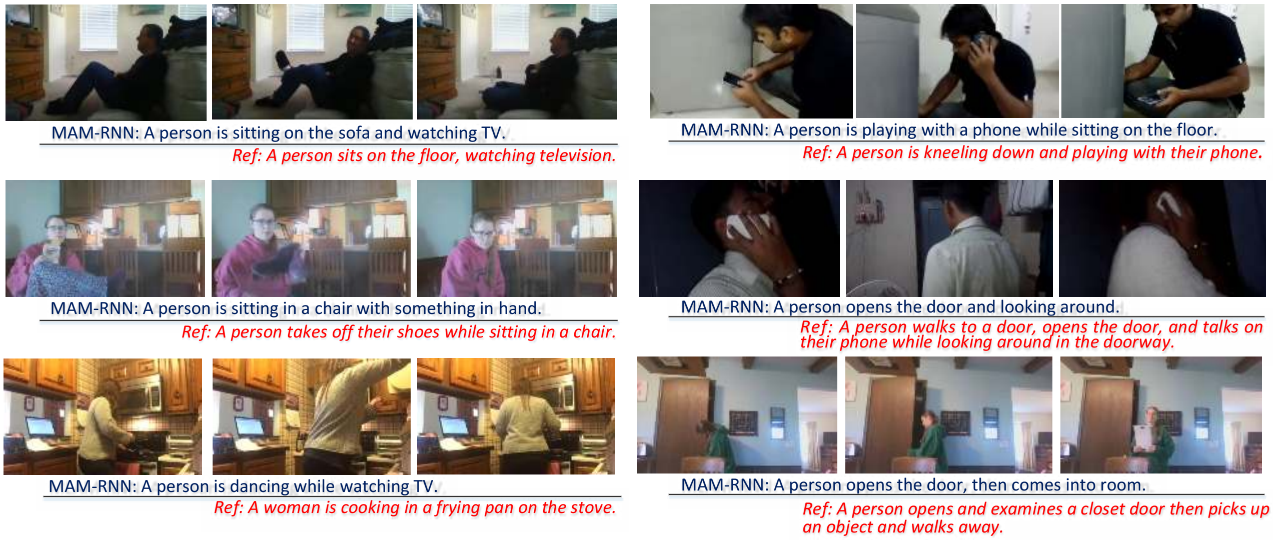
Results on the Charades Dataset

The videos in Charades dataset are divided into three parts, i.e., 7569 for training, 400 for validation and 1863 for testing. Table 3 shows the results of various approaches on the Charades dataset. Generally, S2VT and SA have been introduced before, MAAM is a Memory-Augmented Attention Model which utilizes the memory of past attention when determining the attention weight of current time. Actually, this idea is also employed in our frame-level attention layer. It can be observed in Table 3 that this measure is effective for the task of video captioning, since MAAM outperforms SA (a simple attention model). However, our approach gets even better results, because our multi-level attention model considers the feature jointly in frame-level and region-level, but MAAM is just a frame-level attention model. Besides, the results of our approach with three different features also indicate that our approach is influenced by the extracted features. In other words, given more powerful feature extractor, our approach can get even better performance.

For a better understanding of the results, in Figure 4,

Table 3: The results of various approaches on Charades dataset. (The scores in bold indicate the best value.)

Metrics	CIDEr	BLEU–1	BLEU–2	BLEU–3	BLEU–4	METEOR
S2VT [Venugopalan *et al.*, 2015a]	0.140	0.490	0.300	0.180	0.110	0.160
SA [Yao *et al.*, 2015]	0.181	0.403	0.247	0.155	0.076	0.143
MAAM [Fakoor *et al.*, 2016]	0.167	0.500	0.311	0.188	0.115	0.176
MAM-RNN (VggNet)	0.177	**0.530**	0.307	0.185	0.130	0.179
MAM-RNN (C3D)	0.174	0.509	0.309	0.198	**0.133**	0.183
MAM-RNN (GoogLeNet)	**0.183**	0.506	**0.317**	**0.213**	0.127	**0.191**



MAM-RNN: A person is sitting on the sofa and watching TV.

Ref: A person sits on the floor, watching television.

MAM-RNN: A person is playing with a phone while sitting on the floor.

Ref: A person is kneeling down and playing with their phone.

MAM-RNN: A person is sitting in a chair with something in hand.

Ref: A person takes off their shoes while sitting in a chair.

MAM-RNN: A person opens the door and looking around.

Ref: A person walks to a door, opens the door, and talks on their phone while looking around in the doorway.

MAM-RNN: A person is dancing while watching TV.

Ref: A woman is cooking in a frying pan on the stove.

MAM-RNN: A person opens the door, then comes into room.

Ref: A person opens and examines a closet door then picks up an object and walks away.

Figure 4: Example results of our MAM-RNN on the Charades dataset. Specifically, the blue sentence is generated by MAM-RNN, while the red sentence denotes the reference caption generated by human. As aforementioned, in the Charades dataset, each video has several reference captions. Here, only the reference most similar with the blue one is exhibited.

we provide some examples of the videos and their generated/reference captions of the Charades dataset. As aforementioned, the Charades dataset is much challenging, since each video captures a series of activities of a person. In Figure 4, compared with the reference sentences, we can see that the captions generated by MAM-RNN capture most of the activities. Furthermore, according to the video frames displayed in Figure 4, most of the captions generated by our approach can describe the video content accurately. However, there are also some omissions and even mistakes in the generated captions. It is because that those activities are finished in a very short time, or even invisible, which increases the difficulty for our approach to capture the corresponding visual information. In the future works, we plan to improve our approach and try to solve this problem.

5 Conclusion

In this paper, to extract more fine-grained visual feature for the task of video captioning, we propose a new approach, i.e., MAM-RNN. Specifically, MAM (Multi-level Attention Model) contains two layers, i.e., frame-level attention layer and region-level attention layer, which is employed to encode the most correlated visual feature as the input to the RNN. Then, the RNN is utilized to generate the video caption word by word. The excellent performance on two popular datasets, i.e., MSVD and Charades, has verified that 1) our approach can jointly attend to the salient regions in each frame and the frames correlated to the target caption. 2) our approach can efficiently reduce the interference caused by the irrelevant and meaningless visual feature, and have shown great advantages in improving the accuracy of the video caption.

Last but not least, although our Multi-level Attention Model (MAM) is proposed for video caption in this paper, it is actually a general video feature encoding approach, which can also be used in many other video analogy tasks, such as video classification, summarization, action recognition and so on.

References

[Bengio *et al.*, 1994] Yoshua Bengio, Patrice Y. Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2):157–166, 1994.

[Denkowski and Lavie, 2014] Michael J. Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, 2014.

[Fakoor *et al.*, 2016] Rasool Fakoor, Abdel-rahman Mohamed, Margaret Mitchell, Sing Bing Kang, and Pushmeet Kohli. Memory-augmented attention modelling for videos. *CoRR*, abs/1611.02261, 2016.

[Guadarrama *et al.*, 2013] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *IEEE International Conference on Computer Vision*, pages 2712–2719, 2013.

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

[Krishnamoorthy *et al.*, 2013] Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond J. Mooney, Kate Saenko, and Sergio Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1106–1114, 2012.

[Lin and Och, 2004] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 605–612, 2004.

[Pan *et al.*, 2016] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4594–4602, 2016.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[Shetty and Laaksonen, 2015] Rakshith Shetty and Jorma Laaksonen. Video captioning with recurrent networks based on frame- and video-level features and visual content classification. *CoRR*, abs/1512.02949, 2015.

[Sigurdsson *et al.*, 2016] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526, 2016.

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

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[Thomason *et al.*, 2014] Jesse Thomason, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Raymond J. Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *International Conference on Computational Linguistics*, pages 1218–1227, 2014.

[Vedantam *et al.*, 2015] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015.

[Venugopalan *et al.*, 2015a] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond J. Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence - video to text. In *IEEE International Conference on Computer Vision*, pages 4534–4542, 2015.

[Venugopalan *et al.*, 2015b] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond J. Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1494–1504, 2015.

[Yao *et al.*, 2015] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher J. Pal, Hugo Larochelle, and Aaron C. Courville. Describing videos by exploiting temporal structure. In *IEEE International Conference on Computer Vision*, pages 4507–4515, 2015.

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

[Zeng *et al.*, 2016] Kuo-Hao Zeng, Tseng-Hung Chen, Juan Carlos Niebles, and Min Sun. Title generation for user generated videos. In *European Conference on Computer Vision*, pages 609–625, 2016.