Semantic Enhanced Encoder-Decoder Network (SEN) for Video Captioning

Yuling Gui, Dan Guo, Ye Zhao
School of Computer Science and Information Engineering, Hefei University of Technology
aislinggui@gmail.com,guodan@hfut.edu.cn,zhaoye@hfut.edu.cn

ABSTRACT

Video captioning is a challenging problem in neural networks, computer vision, and natural language processing. It aims to translate a given video into a sequence of words which can be understood by humans. The dynamic information in videos and the complexity in linguistic cause the difficulty of this task. This paper proposes a semantic enhanced encoder-decoder network to tackle this problem. To explore a more abundant variety of video information, it implements a three path fusion strategy in the encoder side which combines complementary features. In the decoding stage, the model adopts an attention mechanism to consider the different contributions of the fused features. In both the encoder and decoder side, the video information is well obtained. Furthermore, we use the idea of reinforcement learning to calculate rewards based on semantic designed computation. Experimental results on Microsoft Video Description Corpus (MSVD) dataset show the effectiveness of the proposed approach.

CCS CONCEPTS

• Human-centered computing → Empirical studies in HCI; **• Computing methodologies → Natural language generation**.

KEYWORDS

Video Captioning; Multi-Feature Fusion; Reinforcement Learning

ACM Reference Format:
Yuling Gui, Dan Guo, Ye Zhao. 2019. Semantic Enhanced Encoder-Decoder Network (SEN) for Video Captioning . In *2nd Workshop on Multimedia for Accessible Human Computer Interfaces (MAHCI '19), October 25, 2019, Nice, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3347319.3356839

1 INTRODUCTION

Video captioning is a complex problem in machine learning combining computer vision and linguistics, which attracts increasing attention. It plays a vital role in a wide range of potential application fields. For healthy people, comprehending the video content is a simple task, whereas it is extremely difficult for individuals who are blind. Audio-described film is a fantastic service to assist individuals with sight loss to understand the video. It uses a recorded narrator to explain the video content. Video captioning is a committed procedure to produce the narrator, which firstly generates text information for speech generation. Video captioning is also widely used in other popular fields such as video retrieval, human-robot interaction. It can be seen that the research on video captioning makes extraordinary contributions to the Human-Computer Interface (HCI) for many common purposes. The earliest captioning task is image captioning [6, 15, 33], aiming at translating an image into the sentence, which is relatively mature nowadays. Due to the complexity, the temporality and the mismatch of video frames and each description word, video captioning faces more challenges than image captioning.

The approach of video captioning can be roughly divided into template-based language methods [11, 27, 35] and sequence to sequence methods [22, 37]. Recently most video captioning methods are based on encoder-decoder architecture, which is an effective approach of sequence training [4, 17, 19]. In the encoder side, features are always extracted by powerful pre-trained convolution neural networks(CNNs), utilizing a fixed-length vector to represent a given video. Different from images, videos are composed of great quantities of video frames, which contains not only static information but also dynamic information. [4] employs two types of feature, appearance feature and motion feature. The appearance feature is the assemble of features of every single frame, and the motion feature is obtained from a feature collection of several video clips. [34] uses a global feature, which describes the information of the whole video. Our proposed model considers all three types of feature, which contains more complementary video information. Recurrent Neural Networks (RNNs) are demonstrated to be suitable for solving sequence problems. After obtaining the visual features of the video, RNNs are usually used to further encode the feature into a vector. The decoder side also uses RNNs to handle the sequence of words.

To consider the visual information dynamically, attention mechanism is proposed, which can pay attention to the key regions in visual processing and achieve remarkable results. A variety of attention mechanisms [2, 21, 37, 39] are employed for video captioning. In the encoder side, an attention mechanism is always implemented to focus on the most relevant spatial regions. In the decoding stage, the attention mechanism can be employed in every frame at each time step [38]. Our SEN adopts an attention mechanism in the decoder side for video frames, which is a static vector. The attention method allows salient video frames to be input in a dynamic way. It assigns different weights to each temporal vector, the less important vectors gain less attention instead of being directly abandoned, which will not cause the loss of potentially important data.

Sequence problems can be considered as a policy-based reinforcement learning task. It allows optimizing the gradient of the

MAHCI '19, October 25, 2019, Nice, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6912-1/19/10…$15.00
https://doi.org/10.1145/3347319.3356839

expected reward, which contributes to image and video captioning tasks [24, 26]. Traditional training methods in video captioning are usually trained by a cross-entropy loss, which aims to maximize the likelihood of the next word by the generating words. As discussed in [26], this method has two drawbacks. At training time, the captioning model generates the next word given the previous ground-truth, whereas at test time it utilizes words sampled by the model. This exposure bias [25] leads to the deviation at test time. Another drawback is that there is no direct relation between loss and the evaluation score. The policy-gradient reinforcement learning method can directly optimize the evaluation result through a reward at test time and tackle the exposure bias problem. [24] proposed a simple weighted cross-entropy scheme for video captioning. Different from traditional training way, it combines the advantages of cross-entropy training and reinforcement training and performs best in all their proposed approaches.

In this paper, we propose a sequence to sequence network to tackle the problem of video captioning. The model obtains effective video information in both encoder and decoder side and adopts a training method of reinforcement learning. The overview of our framework can be seen in Figure 1. The main contributions are summarized as follows:

- The model adopts a strategy of multi-feature fusion in the encoder side. The static, dynamic, and global information are combined in a complementary way in the encoder side, which achieves a great complementary effect.
- It employs an attention mechanism in the decoder side, which pays attention to the important frames while generating every word and uses a method of reinforcement learning, which can directly optimize the non-differentiable metrics.
- The evaluation results on MSVD dataset and the contrast experiments both demonstrate the effectiveness of the model.

The following chapters are distributed as follows. The second part summarizes the principal methods of recent video captioning. The third part introduces the detailed method of our proposed model. The fourth part analyses the experimental results. The last part summarizes the whole work.

2 RELATED WORK

Video Captioning. In the early years, researchers are concentrated on image captioning. Due to the wide application of video correlation technology, video captioning begins to gain more attention in recent years. In the past, [11, 16] uses a two-stage pipeline for video captioning. The first step is to identify semantic information (e.g., subject, verb, object) by detecting the word, and the second step is to generate the word sequence. The template-based approach is difficult to attach the rich language content in human language so that it fails to make a satisfactory description. Inspired by the significant effect of the encoder-decoder framework in machine translation [7], the encoder-decoder framework is also applied in image captioning tasks [9, 33]. [31] employs a sequence to sequence method to tackle the video captioning, which is also a framework of encoder-decoder. In this approach, the sequential frames are encoded firstly and the word is generated one by one. The encoder is generally the CNNs, using the final full connection layer or convolutional layer features as the image features. In video captioning tasks, the encoder uses RNNs to further encode the extracted visual feature, and generate the representation of the video. The decoder is generally composed of the RNNs, which generates a sequence of words that compose the caption of video at each time step. Currently, most captioning methods are based on encoder-decoder architectural.

Feature Fusion. The visual feature contains a variety of information in images. Considering the different categories of the feature, fusing different kinds of features helps to obtain more complete visual information. The multi-feature-fusion strategy has been played a vitally important role in computer vision and is widely used in complex visual tasks [13, 40]. Feature fusion currently is also popular in image and video captioning tasks [8, 34]. Particularly, video data consists of complementary multi-type cues due to a large number of frames and the temporality of video. Nowadays, 3D convolutional neural network (3D CNN) [14] is always used to obtain video features by a set of video frame clips, which is a different way from image feature extractors. [4] employs two strategies to extract the video feature. The first strategy is extracting the appearance feature by an ordinary image feature extractor. The second strategy is using a video feature extractor to obtain the motion feature. It further fuses these two kinds of video features, which subtly combines static and dynamic information. Especially, [34] exploit a diverse video feature fusion, including a global feature to represent the whole video, which is an important feature complement.

Attention Mechanism. The attention mechanism is currently popular in deep learning tasks. It has got extraordinary achievements in machine translation [20], visual captioning [37] and question answering [36]. The attention network is designed to choose the most relevant information to generate the outcome. [33] introduces two attention-based image caption generators under the encoder-decoder framework. One is stochastic hard attention, which adopts the method of object detection and is trained by maximizing the approximate variational lower bound. Another is soft attention, which is trained by back-propagation methods. The effect was significantly improved than the model without attention. [1] proposed bottom-up and top-down attention for captioning and visual question answering, which is a combined attention mechanism. A variety of attention models in video captioning has been used flexibly later [2, 37, 38], which not simply pays attention to the spatial area in an image frame. In the decoder side, since not all frames in a video are equally relevant to the videos, the attention mechanism calculates a weight distribution for the frames.

Reinforcement Learning. Most captioning models are trained by the cross entropy method and maximum likelihood estimation, which aims to maximize the likelihood of the next predicting word. There exists a mismatch problem that the objective function is not the real metric used to evaluate the outcome. It has been shown that reinforcement learning [28] is desirable in the deep end-to-end problem. [25] utilized the reinforcement method to optimize non-differentiable metrics. Reinforcement learning allows optimizing the gradient of the expected reward by sampling from the model while training. [25] uses the reinforcement training to optimize the sequence metrics which is not differentiable. Inspired by the successful application of reinforcement learning in sequence to sequence tasks, the algorithm of SCST [26] for image captioning was proposed. SCST is a reinforcement algorithm for image captioning, which utilizes the evaluation metric score at test time as the reward.

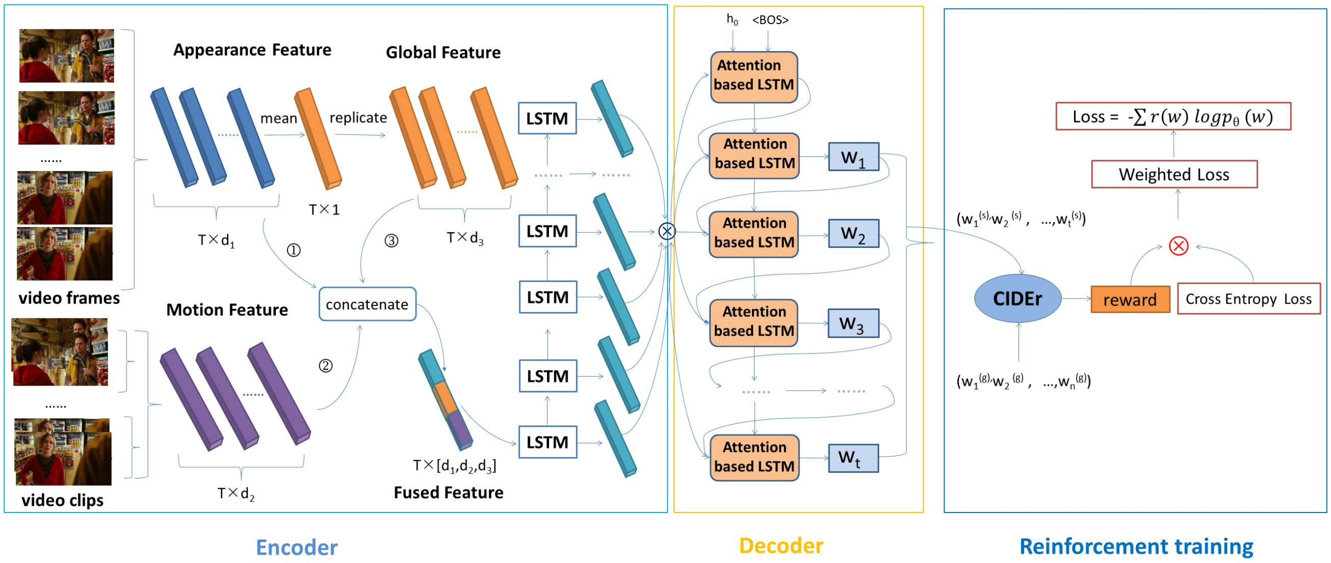


Figure 1: Overview of the proposed model. Given video frames sampled by a video, the appearance feature, action feature and the global feature are produced by different ways. The model utilize an attentive decoder and use the score of CIDEr metric as the reward in reinforcement training. The T represents the time dimension, d represents the feature dimension. Noting that the d_1 equals d_3.

The outperform system are given higher weight and the inferior system are given lower weight. Specifically, [24] proposed a set of reinforcement training methods for video captioning, including a simple weighted cross-entropy scheme for video captioning, which is also a variant of the reinforcement algorithm and achieve great results.

3 METHOD

3.1 Overview of our framework

Video captioning aims at generating a human-understandable sentence to describe a given video. Let $\boldsymbol{V}$ denotes the given video, $[w_1, w_2 \ldots, w_t]$ denotes the sequence of the words. As shown in Figure 1, our proposed model adopts the encoder-decoder framework. First, we employ a triple-path fusion strategy to transform the given video $\boldsymbol{V}$ into the fused feature $\boldsymbol{F}$. The fused feature $\boldsymbol{F}$ is further encoded into the video representation vector $\boldsymbol{Z}$ by a Long Short Term Memory (LSTM) network. Second, the decoder inputs $\boldsymbol{V}$ to an attentive LSTM and generates a sequence of words $[w_1, w_2 \ldots, w_t]$. Finally, we use the CIDEr metric to calculate the reward of each generating sentence and employ the reinforcement training method.

3.2 Multi-Feature-Fusion Encoder

The encoder obtains three types of visual feature and employs a fusion strategy to combine these features. The three features we adopt are the appearance feature, the motion feature and the global feature. The appearance feature represents the static information, the motion feature represents the static information. Particularly, we use the global feature to enhance the local feature learning. Let $\boldsymbol{f} = [f_1, f_2 \ldots, f_n]$ denotes the final feature, $\boldsymbol{f}^{(a)} = [f_1{}^{(a)}, f_2{}^{(a)} \ldots, f_n{}^{(a)}]$ denotes the appearance feature, $\boldsymbol{f}^{(m)} = [f_1{}^{(m)}, f_2{}^{(m)} \ldots, f_n{}^{(m)}]$ denotes the motion feature, and let $\boldsymbol{f}^{(g)} =[f_1{}^{(g)}, f_2{}^{(g)} \ldots, f_n{}^{(g)}]$ denotes the global feature. First, we employ a pre-trained Resnet-50 model [12] to extract the appearance feature $\boldsymbol{f}^{(a)}$ of each video frames. Second, we employ a 3D-CNN model to extract the motion feature $\boldsymbol{f}^{(m)}$. And then the global feature $\boldsymbol{f}^{(g)}$ is obtained by a mean-pooling operation on the extracted appearance feature:

$$\boldsymbol{f}^{(g)} = \frac{1}{n}\sum_{i=1}^{n} f_i{}^{(m)} \tag{1}$$

We fuse the three feature via concatenation operation:

$$\boldsymbol{f} = [\boldsymbol{f}^{(a)}, \boldsymbol{f}^{(m)}, \boldsymbol{f}^{(g)}] \tag{2}$$

The fused feature flexibly complements different kinds of feature. After obtaining the final feature f, the module inputs f into an LSTM to encode the feature into a vector as the representation of the whole video. We use the LSTM architectures following [10], the equation are as follows:

$$\begin{cases} i_t = \sigma(W_{if} f_t + W_{ih} h_{t-1} + b_i) \\ f_t = \sigma(W_{ff} f_t + W_{fh} h_{t-1} + b_f) \\ g_t = \sigma(W_{gf} f_t + W_{gh} h_{t-1} + b_g) \\ o_t = \phi(W_{if} f_t + W_{fh} h_{t-1} + b_f) \\ c_t = f_t \odot c_{t-1} + i_t \odot g_t \\ h_t = o_t \odot \phi(c_t) \end{cases} \tag{3}$$

In the above equation, $\odot$ represents the element-wise product operation, σ represents the sigmoid function, ϕ represents the hyperbolic tangent *tanh*, W_* represents the trained weight matrices, b_* represents the trained biases vectors.

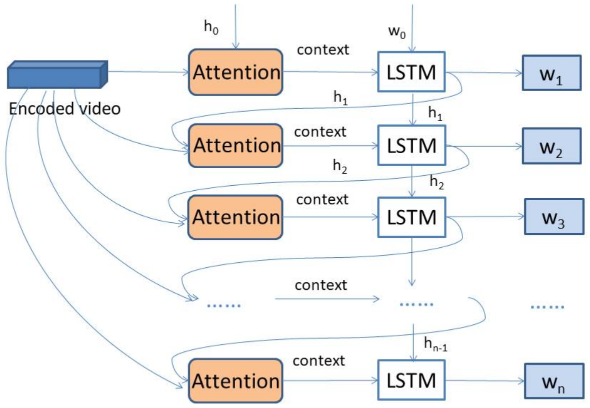


Figure 2: The expanded view of attentive decoder, the context is the weighted vector calculated by attention mechanism.

The number of encoding steps equals the maximum number of frames. The final representation vector Z is an assemble of h_t in each encode step. Assuming the number of encoding step is m, $Z = [z_1, z_2 \dots, z_m]$, which represents the video information.

3.3 Attention-based Decoder

The overview of the specific implementation method of our attention mechanism is shown in Figure 2. Let $z = [z_1, z_2, \dots, z_m]$ denotes the encoded frames. The decoder uses another LSTM to generate the sequence of words. Meanwhile, an attention model is applied in this period. Due to the misalignment of video frames and every word in the caption, the encoded video that should be paid attention to while generating each word. Thus we adopt the attention mechanism in [38], which aims to pay attention to the relevant feature vectors at each time step instead of the spatial area in each frame. We follow [33] to calculate weights and generate context vectors in the attention mechanism. Let the h_t denotes the hidden state in the decode process. The attention module maps the z and the h_t into the same dimension, then puts them into a softmax layer to calculate weights $[\omega_1, \omega_2, \dots, \omega_m]$ at each decode step. Notice that $\sum_{i=1}^{m} \omega_i = 1$.

$$\alpha = Linear(z) \tag{4}$$

$$\beta = Linear(h_t) \tag{5}$$

$$\chi = Linear(\alpha + \beta) \tag{6}$$

$$\omega = softmax(\chi) \tag{7}$$

$$context = R \cdot \omega \tag{8}$$

The encoded vector is weighted as a context vector and be put into the LSTM network with the word generating in the last step. Let w_{t-1} denotes the word vector generating in the last step, h_{t-1} denotes the hidden state generating in the last step, x_t denotes the vector input in next step, which is calculated through an add operation:

$$x_t = w_{t-1} + context \tag{9}$$

$$h_t = lstm(x_t + h_{t-1}) \tag{10}$$

3.4 Reinforcement Training

Reinforcement training is a popular algorithm in machine learning. It is usually categorized as the policy gradient algorithm and Q-learning algorithm. Sequence problems can be considered as a policy-based reinforcement learning task. The decoder can be considered as an agent, the generated words and video features can be considered as the environment, and the prediction of the next word state can be regarded as the action. The score calculated by the evaluation methods represents the reward.

A weighted cross-entropy scheme in [30] was proposed which takes advantage of reinforcement training. Let p_θ denotes the policy gradient, where p_θ represents the parameters in our model. Let p_θ denotes the model, which can be seen as a policy network. Let $w = (w_1, w_2 \dots, w_t)$ denotes the word sequence, where w_t is the word sampled by the model at time t. The aim of reinforcement training is to minimize the negative reward:

$$L(\theta) = -\mathbb{E}_{w\sim p_\theta}[r(w)] = -\sum_{w} r(w) p_\theta(w) \tag{11}$$

To minimize the negative expected reward, $L(\theta)$ might be differentiated:

$$\nabla_\theta \, \mathbb{E}_{w\sim p_\theta}[r(w)] = \mathbb{E}_{w\sim p_\theta}[\nabla_\theta r(w)] \tag{12}$$

In the above equation, $r(w)$ is a discontinuous function relevant to θ, so it is not differential. The equation can be further rewritten as:

$$\begin{aligned} \nabla_\theta \mathbb{E}_{w\sim p_\theta}[r(w)] &= \nabla_\theta \sum_{w} r(w) p_\theta(w) \\ &= \sum_{w} r(w) \, \nabla_\theta \, p_\theta(w) \\ &= \sum_{w} r(w) \, \nabla_\theta \, p_\theta(w) \frac{\nabla_\theta p_\theta(w)}{p_\theta(w)} \\ &= \mathbb{E}_{w\sim p_\theta}[r(w) \, \nabla_\theta \, log p_\theta(w)] \end{aligned} \tag{13}$$

The above equation demonstrates that the $r(w)$ need not be differentiated. [26] proved that choosing the evaluation outcome of CIDEr as a reward outperforms other metrics as a reward. In this work, we follow this rule to choose CIDEr metric as our evaluation method to measure the reward of the generated caption:

$$r(w) = CIDEr(w_1, w_2 \dots, w_n) \tag{14}$$

where w is the word sampled from our model. The loss function is computed as follows:

$$L(\theta) = -\mathbb{E}_{w\sim p_\theta}[r(w) \, \nabla_\theta \, log p_\theta(w)] \tag{15}$$

The weighted loss increases the likelihood of models that gain greater rewards to increase their effect on training results and generate more classical captions. This is different from traditional unweighted cross-entropy models, which often cause heavily penalization while failing to generate results that are not classical in the training data. At test time, we utilize the weighted loss above to optimize our model.

4 EXPERIMENTS

4.1 Datasets

We choose the Microsoft Video Description Corpus (MSVD) [5] as our dataset. MSVD is an assemble of Youtube clips that gathered by Mechanical Turk, which requests people to select short clips describing a single short activity. The annotators give descriptions of these chosen videos and each video is labelled by 40 captions. The description in the corpus is multilingual. In this paper, we choose only the English descriptions and the punctuation is removed. The dataset contains 2,089 Youtube video clips, which are labelled with 85K English descriptions. It principally consists of short videos containing a single action, of which the average duration is 10 seconds. Following previous work [4, 11, 32], we split the dataset by index as follows: 0∼1,199 for training, 1,200∼ 1,299 for validating, 1,299∼ 1,970 for testing.

4.2 Metrics

We employ four metrics: BLEU [23], ROUGE$_L$ [18], CIDEr [30] and METEOR [3] for evaluating the performance of our model. BLEU is the method of word n-grams between generating words and ground-truth words. Following most typical works, we employ a four-grams method, also known as BLEU@4 to evaluate our model effect. ROUGE$_L$ is a method that calculates an F-measure with the recall bias utilizing the way of the longest common subsequence calculation. CIDEr is a method that calculates the mean cosine similarity of n-grams in the generated sentence and the reference captions. It employs TF-IDF to weight them. METEOR aligns the caption with one or more labels. Alignment is based on exact, stem, synonym, phrase, and meaning matches between words or phrases. CIDEr and METEOR have higher accuracy than the other two metrics most of the time, especially while the reference descriptions are fewer. Specifically, we choose the metric CIDEr as the method to further calculate the reward. All metrics are computed using the tool released by Microsoft COCO caption Evaluation Server[1].

4.3 Experiment Setup

Preprocessing and Training Details. The videos are sampled every 9 or 10 frames. We set the maximum length of the image sequence to be 30, and the maximum length of the text sequence to be 30. The length of the clip list is 30, and the length of every clip is 16. In the training process, we use a tag <bos> as the begin of the caption, a tag <eos> as the end of the caption, and a tag <uk> to represent the unknown word. We initialize the cell gate and the hidden state value to be zero in the two LSTMs in our model.

We set the teacher_forcing_ratio to 0.96 at the begin and decrease it to 0.6 gradually. In the probability of teacher_forcing_ratio, ground_truth is used for training. While a new word is generated, the former word in ground truth is used as the input. Otherwise, the model chooses the sampled word as input. The cross-entropy training process is performed by minimizing the log-likelihood loss. We choose the Adadelta optimizer to optimize the parameters of the model. In the loss section, we use doubly stochastic attention regularization the same in [33]. The learning rate is 3e-4 and the decay parameter is ρ=0.95, ϵ=10×10^{-7}. To prevent overfitting, we dynamically reduced the learning rate after 50 epochs. The model is trained for 100 epochs in this stage. In the reinforcement training period, the model uses the CIDER metric as the reward weighted the cross-entropy loss. We train the model for one epoch in this stage on the premise of completing cross-entropy training.

Table 1: Experimental Results on MSVD, where CIDEr-D, B@4, ROUGE$_L$ and METEOR are different metrics for evaluation. For our SEN model, A represents using the appearance feature, M represents using the motion feature, G represents using the global feature, a represents using the attention mechanism, and rl represents using the reinforcement training.

Model	CIDEr-D	B@4	ROUGE$_L$	METEOR
LSTM-YT [32]	-	33.3	-	29.1
SA-Googlenet-C3D [37]	-	41.9	-	29.6
S2VT-VGG+Flow [31]	-	-	-	29.8
BANET [4]	63.5	42.5	-	**32.4**
LSTM-E [22]	-	45.3	-	31.0
SEN (A-M)	57.7	40.7	66.8	29.3
SEN (A-M-G)	59.6	40.5	65.8	29.4
SEN (A-M-a)	62.7	42.5	67.6	30.3
SEN (A-M-G-a)	64.7	42.7	68.0	30.0
SEN (A-M-a-rl)	65.6	46.2	66.8	30.8
SEN (A-M-G-a-rl)	**67.0**	**46.5**	**69.3**	31.6

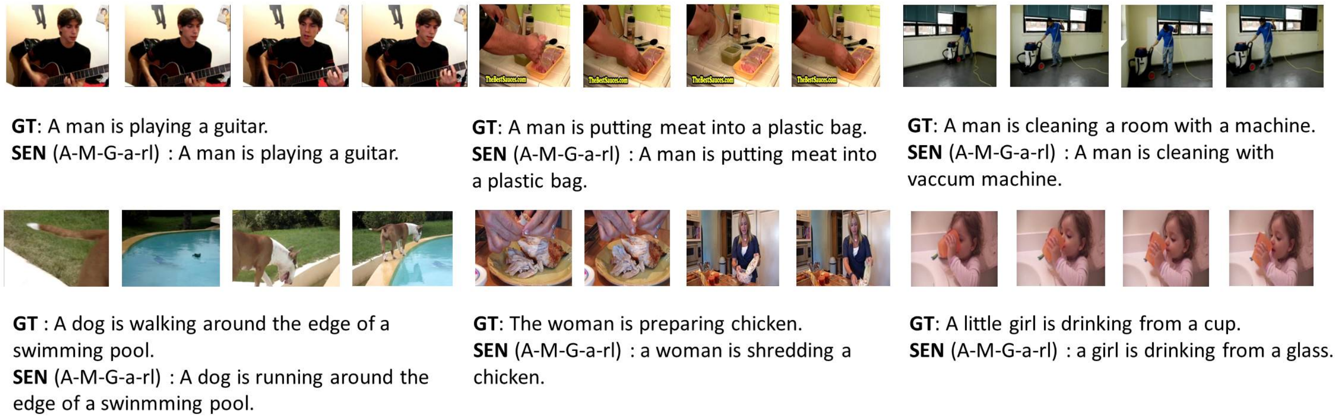
Feature Extraction Process. The appearance feature means static information, which requires a network for extracting image features. In this work, for the appearance feature, we employ a pre-trained Resnet-50 model, to extract features for every sampled frame. The Resnet-50 model is pre-trained on the ImageNet dataset. We apply the activation units at the penultimate layer of the Resnet-50 model and the dimension of the motion feature is 2,048. For motion appearance, we firstly generate video clips following the method mentioned in preprocessing. We further input every video clip into a 3D-CNN network [29]. The 3D-CNN model is pre-trained on the Sports-1M dataset. Similar to obtaining a motion feature, we also apply the activation units at the penultimate layer of the 3D-CNN model. The dimension of the motion feature is 4,096. As introduced in the method part, for global features, we implement a mean-pooling operation on the produced appearance feature. In order to keep the dimensions consistent, we make 2,048 replications for the acquired features. The dimension of the global feature is consistent with the appearance feature.

Time Complexity. In our experiment, it takes 40 minutes to extract 1,300 video features. When the training batch-size is set to 30, the entire training process is seven hours. The trained model can generate the features of video into text immediately. For a single given video in our dataset, it takes less than three seconds to process the video and extract all the features. It can be seen that this algorithm has a promising prospect for real-time application such as provide service for audio described films.

4.4 Experimental Results

On the MSVD dataset, we carry out a number of comparative experiments. Our semantic enhanced encoder-decoder network is abbreviated as SEN. The experimental result is shown in Table 1. Here, A represents using appearance feature, M represents using the motion feature, G represents using the global feature, a represents using attention mechanism, and rl represents using reinforcement learning.

[1] https://github.com/tylin/coco-caption



GT: A man is playing a guitar.
SEN (A-M-G-a-rl) : A man is playing a guitar.

GT: A man is putting meat into a plastic bag.
SEN (A-M-G-a-rl) : A man is putting meat into a plastic bag.

GT: A man is cleaning a room with a machine.
SEN (A-M-G-a-rl) : A man is cleaning with vaccum machine.

GT : A dog is walking around the edge of a swimming pool.
SEN (A-M-G-a-rl) : A dog is running around the edge of a swinmming pool.

GT: The woman is preparing chicken.
SEN (A-M-G-a-rl) : a woman is shredding a chicken.

GT: A little girl is drinking from a cup.
SEN (A-M-G-a-rl) : a girl is drinking from a glass.

Figure 3: Comparison of our example results and ground-truth. Here, GT means ground-truth and SEN (A-M-G-a-rl) represents our methods. A represents using appearance feature, M represents using the motion feature, G represents using the global feature, a represents using the attention mechanism, and rl represents using the reinforcement learning.

Table 2: The comparative experiment of SEM (A-M), SEM (A-M-G), SEN (A-M-a) and SEN (A-M-G-a), G represents using the global feature, a represents using the attention mechanism, which shows the role of feature fusion and attention mechanism.

Model	CIDEr	B@4	ROUGE$_L$	METEOR
SEN (A-M)	57.7	40.7	66.8	29.3
SEN (A-M-G)	59.6	40.5	65.8	29.4
SEN (A-M-a)	62.7	42.5	67.6	30.3
SEN (A-M-G-a)	64.7	42.7	68.0	30.0

Table 3: The comparative experiment of SEN (A-M-a), SEN (A-M-a-rl), SEN (A-M-G-a) and SEN (A-M-G-a-rl), where rl represents using the reinforcement learning, which shows the role of reinforcement learning.

Model	CIDEr	B@4	ROUGE$_L$	METEOR
SEN (A-M-a)	62.7	42.5	67.6	30.3
SEN (A-M-a-rl)	65.6	46.2	66.8	30.8
SEN (A-M-G-a)	64.7	42.7	68.0	30.0
SEN (A-M-G-a-rl)	67.0	46.5	69.3	31.6

Table 4: The comparative experiment of our models of different components of modules.

Model	CIDEr	B@4	ROUGE$_L$	METEOR
SEN (A-M)	57.7	40.7	66.8	29.3
SEN (A-M-G-a)	64.7	42.7	68.0	30.0
SEN (A-M-a-rl)	65.6	46.2	66.8	30.8
SEN (A-M-G-a-rl)	67.0	46.5	69.3	31.6

Table 5: The contrast experiment of SEN (A-M-G), SEN (A-M-G-a) and SEN (A-M-G-a-rl) with other works.

Model	CIDEr	B@4	ROUGE$_L$	METEOR
LSTM-YT [32]	-	33.3	-	29.1
SEN (A-M-G)	59.6	40.5	65.8	29.4
SA-Googlenet-C3D [37]	-	41.9	-	29.6
S2VT-VGG+Flow [31]	-	-	-	29.8
BANET [4]	63.5	42.5	-	**32.4**
SEN (A-M-G-a)	64.7	42.7	68.0	30.0
LSTM-E [22]	-	45.3	-	31.0
SEN (A-M-G-a-rl)	**67.0**	**46.5**	**69.3**	31.6

First, we compare our own models with different modules. The based framework of our work uses a simple encoder-decoder framework, which adopts an LSTM to encode the video and uses another LSTM unit to generate words. SEN (A-M) represents using the fusion feature of appearance feature and motion feature in the encoder side, which is the most basic model to compare in our work. Furthermore, we carry out contrast experiments with other classic captioning works including LSTM-YT [37], S2VT [31], LSTM-E [22]. Particularly, we compare our work with BANET [3], which employs a hierarchical boundary-aware neural coder for video captioning and uses a boundary detector to enhance the function of the encoder. This model is based on the encoder-decoder framework and adopts the fusion strategy to combine appearance feature and motion feature, which is very similar to what we do in this respect.

It can be seen from the comparative experiments that feature fusion, attention mechanism, and reinforcement training methods all play a good role in our model. The example results of our model are shown in Figure 3. It can be seen that in some cases our model outputs a completely accurate result. In other cases, our model predicts results very close to the ground-truth, which also accurately describes video content.

The effectiveness of feature fusion. It can be seen from Table 2, SEN (A-M-G) adds the global feature to the based SEN (A-M) model, which improves the CIDEr metric from 57.7 to 59.6. SEN

(A-M-G-a) adds the global feature to the A-M-a model, which improves the CIDEr metric by 2 points. In contrast, models with global features are more effective than models without global features. All these two comparisons indicate the global features complement the other two features effectively.

The effectiveness of attention model. The result in Table 2 also shows the significant effect of our attention mechanism. SEN (A-M-a) adds an attention mechanism in the decoder side to the A-M model, which improves the CIDEr metric by 5 points and has a prominent effect on all the other metrics. SEN (A-M-G-a) adds the attention mechanism to the SEN (A-M-G) model, which improves the CIDEr metric by 5.1 points and also has a prominent effect on all the other metrics. Both of these two groups of comparative experiments effectively prove the remarkable effect of attention mechanism at the decoder side.

The effectiveness of reinforcement training. As is shown in Table 3, SEN (A-M-a-rl) adds a reinforcement training method to SEN (A-M-a) model, which improves the CIDEr by 2 points. SEN (A-M-G-a-rl) adds a reinforcement training method to SEN (A-M-G-a) model, which improves the CIDEr by 2.3 points and improves the effect on other metrics meanwhile. These demonstrate that compared to just using cross-entropy, reinforcement training is an ingenious method to optimize the evaluation results.

Comprehensive effect of the model. As shown in Table 4, SEN (A-M-G-a) adds the global feature in the encoder side and the attention mechanism in the decoder side respectively to the based model, which demonstrates the effectiveness of the common effect of feature fusion on the encoding side and attention mechanism on the decoding side. SEN (A-M-a-rl) employs attention mechanism and reinforcement learning method to the A-M model, which demonstrates the common effect of attention mechanism and reinforcement learning. SEN (A-M-G-a-rl) outperforms all our models, showing the co-efficiency of our modules.

Contrast experiment with models in other works. We compare our experimental results with several well-known works. LSTM-YT implements only an appearance feature to encode the video. There has been a marked improvement of our SEN (A-M-G) model in the BLEU$_4$ and METEOR, which demonstrates the effectiveness of our fusion strategy encoder. SA extracts video features by GoogleNet and 3D-CNN network but uses a simple LSTM to decode the video. Compare our SEN (A-M-G-a) model with SA, our score of CIDEr is 5.1 higher than SA, which shows our attention-based decoder does have a marked effect. Our SEN (A-M-G-a) also outperforms the classic sequence model S2VT. The model of BANET uses a structure of hierarchical LSTM and employs a boundary detector, which aims to alter the temporal connections of the network for a given video. The purpose of attention mechanism in our model is to focus on the important frames, which is also a selection operation and somewhat similar to the aim of BANET. In our work, we leave out the part of boundary detection and only adopt one layer of the LSTM structure to simplify the model of the encoder side. SEN (A-M-G-a) exceeds BANET in both CIDEr and BLEU$_4$ metric. LSTM-E exploit a method of visual-semantic embedding for video captioning. Our SEN (A-M-G-a-rl) outperforms the LSTM-E in BLEU$_4$ and METEOR, which employs a more simple encoder.

All the comparative experiments show the remarkable result, especially in the CIDEr metric, all modules in our model show significant function, which demonstrates the effectiveness and robustness of our model.

5 CONCLUSION

In this work, we propose a semantic enhanced encoder-decoder network for video captioning. The encoder first exploits a three-path fusion strategy to effectively leverage three complementary features effectively. The decoder adopts an attention mechanism to consider the different contributions of the fused feature while generating words at each time step. The model further utilizes the idea of reinforcement learning to calculate rewards based on semantic designed computation to optimize the model. The performance on MSVD dataset and the contrast experiments all demonstrate the effectiveness of our model.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (NSFC) under grants 61876058 and 61632007.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6077–6086.

[2] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. 2015. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432* (2015).

[3] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.

[4] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. 2017. Hierarchical boundary-aware neural encoder for video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1657–1666.

[5] David L Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 190–200.

[6] Xinlei Chen and C Lawrence Zitnick. 2014. Learning a recurrent visual representation for image caption generation. *arXiv preprint arXiv:1411.5654* (2014).

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[8] Tien X Dang, Aran Oh, In-Seop Na, and Soo-Hyung Kim. 2019. The Role of Attention Mechanism and Multi-Feature in Image Captioning. In *Proceedings of the 3rd International Conference on Machine Learning and Soft Computing*. ACM, 170–174.

[9] 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*. 2625–2634.

[10] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 6645–6649.

[11] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE international conference on computer vision*. 2712–2719.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[13] Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R Hershey, Tim K Marks, and Kazuhiko Sumi. 2017. Attention-based multimodal fusion for video description. In *Proceedings of the IEEE international conference on computer vision*. 4193–4202.

[14] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2012. 3D convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2012), 221–231.

[15] Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3128–3137.

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

[17] Guang Li, Shubo Ma, and Yahong Han. 2015. Summarization-based video caption via deep neural networks. In *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 1191–1194.

[18] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[19] Yuan Liu and Zhongchao Shi. 2016. Boosting video description generation by explicitly translating from frame-level captions. In *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 631–634.

[20] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[21] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. 2016. Hierarchical recurrent neural encoder for video representation with application to captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1029–1038.

[22] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2016. Jointly modeling embedding and translation to bridge video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4594–4602.

[23] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 311–318.

[24] Sang Phan, Gustav Eje Henter, Yusuke Miyao, and Shin'ichi Satoh. 2017. Consensus-based sequence training for video captioning. *arXiv preprint arXiv:1712.09532* (2017).

[25] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732* (2015).

[26] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7008–7024.

[27] 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 IEEE International Conference on Computer Vision*. 433–440.

[28] Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*. Vol. 2. MIT press Cambridge.

[29] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 4489–4497.

[30] 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*. 4566–4575.

[31] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*. 4534–4542.

[32] 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 preprint arXiv:1412.4729* (2014).

[33] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3156–3164.

[34] Ning Xu, An-An Liu, Yongkang Wong, Yongdong Zhang, Weizhi Nie, Yuting Su, and Mohan Kankanhalli. 2018. Dual-stream recurrent neural network for video captioning. *IEEE Transactions on Circuits and Systems for Video Technology* (2018).

[35] Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. 2015. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[36] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 21–29.

[37] 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 IEEE international conference on computer vision*. 4507–4515.

[38] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. 2015. Video description generation incorporating spatio-temporal features and a soft-attention mechanism. *arXiv preprint arXiv:1502.08029* (2015).

[39] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. 2016. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4584–4593.

[40] Zhenli Zhang, Xiangyu Zhang, Chao Peng, Xiangyang Xue, and Jian Sun. 2018. Exfuse: Enhancing feature fusion for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 269–284.