Syntax-Aware Action Targeting for Video Captioning

Qi Zheng Chaoyue Wang Dacheng Tao
UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering,
The University of Sydney, Darlington, NSW 2008, Australia
{qi.zheng,chaoyue.wang,dacheng.tao}@sydney.edu.au

Abstract

Video captioning aims to describe objects and their interactions in the video using natural language. Existing methods have made great efforts to identify objects in videos, but few of them emphasize the prediction of interactions among objects, which is usually indicated by action/predicate in generated sentences. Different from other components in a sentence, the predicate depends on both the static scene and the dynamic motions in a video. Due to the neglect of such uniqueness, actions generated by existing methods may depend heavily on the co-occurrence of objects, e.g. *'driving' is predicted with high confidence whenever both man and car are detected. In this paper, we propose a Syntax-Aware Action Targeting (SAAT) module that explicitly learns actions by simultaneously referring to the subject and video dynamics. Specifically, we first identify the subject by drawing global dependence among multiple objects, and then decode action from a common space that fuses the embedding of the subject and the temporal feature of the video. Validated on two public datasets, the proposed module increases action accuracy in generated descriptions, which present better semantic consistency with the dynamic content in videos. Codes are available on* `https://github.com/SydCaption/SAAT`.

1. Introduction

The goal of video captioning is to automatically generate a complete and natural sentence to describe video content, ideally encapsulating its most informative dynamics [58, 17, 13]. Such dynamics usually reveal a specific action within the video clip, such as *running*, *eating* and *jumping*. Compared with image captioning [2, 31, 20, 8] that aims at depicting the static scene in an image, video captioning emphasizes more on the action and attracts increasing attention in the field of both computer vision and artificial intelligence. It has extensive applications such as Visual Question Answering (VQA) [57, 16], human-robot interaction [32] and video retrieval [64, 14].

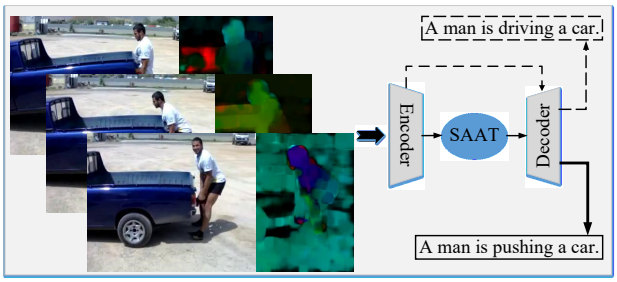

Figure 1: Example of caption generation with and w/o the SAAT module. The dotted line demonstrates captioning using a regular encoder-decoder framework, and the solid line shows the result using an SAAT-involved captioner.

Owing to recent advancements in object recognition using deep learning [42, 41, 18], exciting progress has been made on video captioning. Specifically, a trend in existing methods [35, 66, 65, 1, 56, 59, 6, 55] is that they devise diverse modules to identify the objects in a video clip. This has greatly improved the relevance between the generated captions and the given video due to at least two advantages. On the one hand, with abundant static information extracted from the video clip, a captioner is more likely to depict the targeted *instance* in the video. On the other hand, the co-occurrence of objects helps the captioner to *remember* <video, description> pairs. A clear illustration of this can be found in [38], where captions are generated by recalling similar scenes in learned *memory*.

The downside of these methods is the ignorance of action (*i.e.* the predicate of a sentence) learning, which requires more dynamic information from videos than other components in a sentence. This can hardly be remedied by *memory*. For example, the learned model may depend heavily on what it has seen in the training process such as the prior of co-occurrence to generate captions for a video. As demonstrated in Fig. 1, when *man* and *car* are both detected, regular encoder-decoder framework tends to give *a man is driving a car* even though the man is outside the car and the car is not moving forward. This causes an enormous

divergence between generated descriptions and the original content in videos. Unfortunately, such a divergence can hardly be lessened by minimizing a sentence-level cross-entropy loss since the remaining words can exactly match those in a human-annotated sentence though the *action* is wrong. Either, the generated captions can still achieve high scores w.r.t those automatic metrics such as BLEU [36], METEOR [10] and CIDEr [51] in the case of wrong-action prediction [37].

To this end, we propose a Syntax-Aware Action Targeting (SAAT) model for video captioning in this paper. By counting the loss of visually-related syntax components (*i.e*. *subject*, *object* and *predicate*), we explicitly target actions in video clips to afford a captioner extra guidance apart from linguistic prior. It is worth noting that different from the work [54], where POS (Part-of-Speech) tag of each word is predicted to guide the captioning process, we only focus on the components that convey the most visual information to target action and guide caption generation. Specifically, our model firstly generates scene representation using both regional RGB features and the location of regions by learning a self-attention module [50]. Empirically, we use Faster R-CNN [42] as the object detector to generate region proposals and extract regional features, where other detectors can also be adopted. The learned self-attended representation is expected to draw global dependence among multiple objects within the scene. Then the syntax components *subject*, *object* and *predicate* are decoded from the representation by setting different queries. After targeting the *action*, *i.e*. the *predicate*, an action-guided captioner is devised to generate descriptions for the input video. An attention distribution over the targeted *action* and the previously predicted words is learned to guide the prediction of the next word. The whole model is trained in an end-to-end manner and the objective is to minimize the weighted sum of the loss caused by components prediction and that by caption generation. To summarize, the contributions of this paper are three-fold:

- We propose a syntax-aware module that forms a self-attended scene representation to model the relationship among video objects and then decodes syntax components (*subject*, *object* and *predicate*) by setting different queries, targeting the action in video clips.
- We devise an action-guided captioner that learns an attention distribution to dynamically fuse the information from the *predicate* and previously predicted words, avoiding wrong-action prediction in generated captions.
- Extensive results on benchmark datasets demonstrate the superiority of the proposed method in terms of the automatic metrics BLEU, METEOR, ROUGE and CIDEr. Compared with the regular captioner (*i.e*. the Baseline), the proposed model brings a relative 2.7% and 5.9% increase of CIDEr score on MSVD and MSR-VTT dataset respectively, promoting the quality of generated captions.

2. Related Works

2.1. Video captioning

Most of the early methods for video captioning are based on specific templates such as $\{$*who did what to whom, and where and how they did it*$\}$ [26, 9, 4]. These methods require lots of hand-designed linguistic rules and deal with limited categories of objects, actions or attributes. With the rise of deep neural networks, an encoder-decoder framework was firstly proposed to overcome these limitations in [53]. This framework explores the power of CNNs in video representation and RNNs in sequential learning. On top of that, methods such as conditional random fields (CRF) [12] and recurrent neural networks (RNNs) [12, 52] are proposed to replace the original mean-pooling in the encoder, and attention mechanisms [15, 35, 62, 28] were incorporated to extract salient frames or regions in decoding phase. For instance, Xu *et al.* [60] proposed the widely-used soft-attention (SA) encoder-decoder model. hRNN [63] employs both temporal- and spatial-attention mechanisms during sentence generation and also learns a paragraph generator to capture the inter-sentence dependency. Multimodal features were also exploited. For example, LSTM-E [34] simultaneously explores the learning of LSTM and visual-semantic embedding. DMRM [61] introduces a Dual Memory Recurrent Model to incorporate the temporal structure of global features and regions-of-interest features.

Recent latest works include diverse adjustments to the encoder-decoder framework. M3 [56] builds a visual and textual shared memory to model the long-term visual-textual dependency and further guide visual attention no described visual targets. MA-LSTM [59] exploits both multimodal streams and temporal attention to selectively focus on specific elements during sentence generation. Chen *et al.* [6] proposes a frame picking module (PickNet) to select informative frames from a video. RecNet [55] employs backward flow to reproduce the video features while generating descriptions. OA-BTG [65] captures salient objects with their temporal dynamics. GRU-EVE [1] uses Short Fourier Transform to embed video features and then encodes them using Gated Recurrent Units. Due to the limitation that these methods can hardly build correct correspondence between a word (*e.g*. the action) and video content, MARN [38] exploits a memory structure to explore the full-spectrum correspondence between a word and its various visual contexts.

2.2. Captioning using syntax information

The role of syntax components, which are considered to contain more semantic information, has been emphasized in sentence/text generation [22, 49, 7, 21]. Since captioning is a task involving both natural language processing and computer vision, several attempts that utilize visually-related

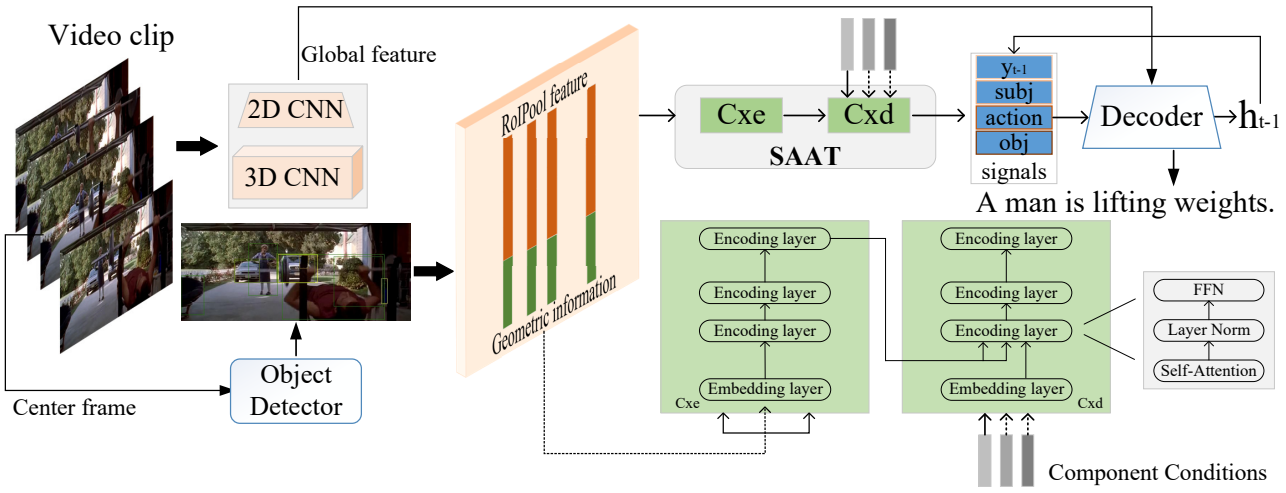

Figure 2: The architecture of our Syntax-Aware Action Targeting (SAAT) involved decoding process. The SAAT module consists of two blocks: 1) Component extractor-encoder (**Cxe**) that computes scene representation based on self-attention of bounding boxes, 2) Component extractor-decoder (**Cxd**) that predicts different syntax components based on their own conditions. 2D-, 3D-CNNs and object detector are used to extract features from the input video clip. A self-attended scene representation is learned by **Cxe** and used to decode syntax components including *subject*, *object* and *predicate* by **Cxd**. Finally, the targeted *action* is exploited to guide the generation of descriptions.

syntax information have been made to image/video captioning. For example, Lebret *et al.* [27] analyzed phrases in captions and suggested learning a common space for image and phrase representations. Tan and Chan [47] proposed a phrase-based hierarchical LSTM model, which is composed of a phrase decoder and an abbreviated sentence decoder. Ling and Fidler [30] explored teaching machines to describe images by correcting mistaken phrases. These methods generally solve a multi-task problem. Beyond them, He *et al.* [19] discovered that the Part-of-Speech (POS) tags of a sentence are effective cues for guiding the Long Short-Term Memory (LSTM) based word generator. Deshpande *et al.* [11] suggested predicting POS as summaries of an image, based on which captions are generated. POS-CG [54] simultaneously learns a POS sequence generator with the description generator. Different from these works, we synthetically utilize syntax information to target action and then guide caption generation.

3. Method

Given a video, our model takes as input multi-modal features $V = \{V^r, V^m, V^b\}$ extracted from the video, *e.g.* RGB feature from 2D CNNs, temporal feature from C3D network and feature of local regions from object detectors, respectively. Our model firstly generates scene representation from the available features by learning a self-attention module, which will be described in Section 3.1. In Section 3.2, we outline how to decode syntax components such as *subject*, *predicate* and *object* from the scene representation, and use the *action* (*i.e. predicate*) to guide caption generation. In Section 3.3, we give details of the training and inference process. The overall framework is shown in Fig. 2.

3.1. Self-attended scene representation

The RGB feature of a video is frame-level, which can be seen as the global context of the video. Features of object regions, on the other hand, provide local information in finer detail. Unlike other approaches that consider the object regions as independent boxes, we desire to learn a representation composed of both their semantic information and spatial location, which is expected to help the model understand the scene.

Inspired by the self-attention mechanism [50] in natural language processing, we design an encoder based on self-attention to draw global dependence among multiple objects within a scene, as show in Fig. 2. Here, the component extractor-encoder **Cxe** maps an input sequence of regional features $V^b = (v_1^b, \ldots, v_K^b)$ to a sequence of continuous representations $V^{b'} = (v_1^{b'}, \ldots, v_K^{b'})$, where K is the number of object regions. Given $V^{b'}$, the component extractor-decoder **Cxd** then generates POS tags, *i.e. subject*, *predicate*, *object*.

According to [50], the scaled dot-product attention of queries Q given <*key, value*> pairs is produced by

$$f_{att}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}}V) \quad (1)$$

where d_k is the dimension of queries and keys. In our case,

the queries, keys and values are all projections of the regional feature. Similar to the importance of sequence order in natural language, spatial location is vital to determine the semantic information conveyed by visual content. Therefore, we add embedding of object location to form a complete representation of a region through concatenation

$$(Q, K, V) = (R_c W^Q, R_c W^K, R_c W^V) \tag{2}$$

where the projections $\{W^Q, W^K, W^V\} \in \mathbb{R}^{d_c \times d_k}$ are parameter matrices to be learned, d_c is the dimension of the input feature, d_k is the unit number of our model, and

$$R_c = \mathrm{ReLU}([W_l^T R_l; W_b^T V^b]) \tag{3}$$

where the $[\cdot;\cdot]$ denotes the concatenation of two matrices, $R_l = [X, Y, W, H] = [r_{l_1}, r_{l_2}, \ldots, r_{l_K}]$ provides information of the center coordinates, width and height of the regions, which is normalized by the size of video frames as $r_{l_i} = [\frac{x_i}{w_f}, \frac{y_i}{h_f}, \frac{w_i}{w_f}, \frac{h_i}{h_f}]^T$, where w_f and h_f are the width and height of a frame in the video. Similar to [50], this module can be easily extended to its multi-head version, which is omitted due to space limitation.

The physical interpretation behind this modeling is that the scene composed of multiple objects is not only determined by the quantity and categories of the objects, but also relates to their spatial arrangement. By embedding the relative position of objects, the learned scene representation $V^{B'}$ is expected to contain the spatial relationship among objects. On top of this, syntax components such as *subject*, *object* and *predicate* can be decoded from the scene since they are more related to the visual scene compared with other components in a sentence.

3.2. Syntax-aware action targeting captioning

We consider that the limitation of existing methods based on the regular encoder-decoder framework [60] is the correspondence between generated actions in a description and the dynamic content in videos. To this end, we overcome this issue by first targeting the action in a video and then use it to guide the captioning process. Intuitively, *subject* and *object* rely more on spatial appearance of regions, and *predicate* requires the temporal information from in video clip. In our view, the word that is predicted to describe the action in a video also depends on the specific *subject*. For example, when the *subject* belongs to animate beings, possible *actions* can be *running, walking, fighting, cooking*, etc; when the *subject* belongs to inanimate objects, action in passive voice is likely to be produced.

To target the action in a video, we first decode the *subject* from the self-attended scene representation given by the former section. As shown in Fig. 3, we set the global RGB

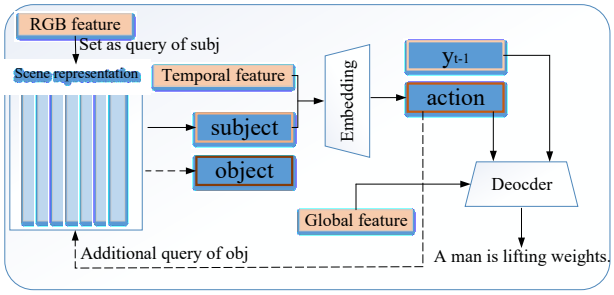

Figure 3: Illustration of decoding syntax components from self-attended scene representation.

feature as the query of the *subject*, which gives

$$s = arg \max_{w \in vocab} p_\theta(w \mid V^{b'}, V^r) \tag{4}$$

$$p_\theta(w \mid V^{b'}, V^r) = \mathrm{softmax}(W_s^T f_{att}(V^{r'}, V^{b'}, V^{b'})) \tag{5}$$

where $V^{r'}$ is the projected global feature of the video, *i.e.* $V^{r'} = W_v^T V^r$, and $V^{b'}$ is the learned scene representation. θ denotes the parameters to be learned.

Then the *predicate* is decoded given the *subject* and the temporal change in the video

$$a = arg \max_{w \in vocab} p_\theta(w \mid s, V^m) \tag{6}$$

$$p_\theta(w \mid s, V^m) = \mathrm{softmax}(W_a^T \mathrm{ReLU}([\mathbf{E}s; V^{m'}])) \tag{7}$$

where s is the predicted *subject*, $V^{m'}$ is the projected motion feature of the video, *i.e.* $V^{m'} = W_m^T V^m$, and $\mathbf{E}$ is the embedding of words in the vocabulary.

Finally, the *object* is decoded given the *predicate* and the scene representation

$$o = arg \max_{w \in vocab} p_\theta(w \mid a, V^{b'}) \tag{8}$$

$$p_\theta(w \mid a, V^{b'}) = \mathrm{softmax}(W_o^T f_{att}(\mathbf{E}o, V^{b'}, V^{b'})) \tag{9}$$

where a is the predicted action, and the embedding of which is set as the query of *object*.

To generate action-relevant descriptions of a video, we devise a syntax-guided captioner that uses the *action* produced by the SAAT module. It is worth noting that the specific guidance passing to the captioner is flexible. We adopt the *action* guided captioner since we observe that most objects in videos can be correctly predicted by a regular decoder. We implement the captioner with LSTMs. To enable the captioner jointly refer to the information from syntax components and the information from previously predicted words, an attention distribution over them is learned

$$\beta_{t,j} = \mathrm{softmax}(v_\beta^T \tanh(W_{\beta h} \mathbf{E} y_j + W_h h_{t-1} + b_\beta)) \tag{10}$$

where t represents the time step as in regular decoders, $y_j \in \{y_a,\ y_{t-1}\}$, j is the corresponding index and $\sum_j \beta_{t,j} = 1$. The probability distribution of word y_t is produced by

$$p_\theta(y_t \mid y_j) = \mathrm{LSTM}(\sum_j \beta_{t,j} \mathbf{E} y_j, W_v \bar{v}, h_{t-1}) \quad (11)$$

where $\bar{v}$ denotes the average of global features V^r and V^m over time-space and W_v is the projection matrix to be learned. y_0 is given by the *bos* token and h_0 is a zero vector.

3.3. Training and inference

The objective of our model is to minimize the sum of loss $\mathcal{L}_s$ from the SAAT module and loss $\mathcal{L}_c$ from the captioner

$$\mathcal{L}(\theta) = \mathcal{L}_c + \lambda \mathcal{L}_s \quad (12)$$

where λ is a hyper-parameter to balance the two terms, and

$$\mathcal{L}_c = -\sum_{i=1}^{N} \sum_{t=1}^{T_i} \log p_\theta(y_t = y_t^* | y_{1:t-1}, y_a) \quad (13)$$

$$\mathcal{L}_s = -\sum_{i=1}^{N} \log p_\theta((s, a, o) = (s^*, a^*, o^*) | V^{[b,r,m]}) \quad (14)$$

where $y^*_{1:T_i}$ are human-annotated captions and $(s^*,\ a^*,\ o^*)$ are syntax components generated by NLP tool[1].

By counting the loss of visually-related syntax components (*i.e*. *subject*, *object* and *predicate*), we explicitly target actions, which require more dynamic information from videos than other components in a sentence, in video clips. The predicted action is then guide the captioning process by affording a captioner extra guidance apart from linguistic prior to generate action-relevant descriptions.

In the training process, the object number K is fixed for all the samples to allow mini-batch training. RGB features and locations of K object regions are extracted from the center frame of each video to learn the scene representation. In our experiments, we set $K = 10$. If more than K objects are detected, the K objects with the highest confidence are selected. If less than K, then some of them will appear more than once, when the location information can be used to distinguish repeated regions. During inference, K can be arbitrary for an input video. We observe that the pre-trained object detector sometimes fails to capture the desired objects from videos, which may be caused by the low-resolution of video frames and the size of objects. Therefore, during learning the syntax-aware scene representation, we add an additional empty region to the K selected object regions to allow the object-missing case.

[1] https://www.nltk.org

4. Experiments

We compare our method with existing ones on two popular benchmark datasets from the literature in video captioning, *i.e*. Microsoft Video Description (MSVD) dataset [17] and MSR-Video To Text (MSR-VTT) dataset [58]. We first give details of the two datasets and preprocessing performed in this work, and then we discuss the experimental results.

4.1. Datasets

Microsoft video description corpus (MSVD). [17] This dataset contains $1,970$ YouTube open domain video clips. Generally, each clip predominantly shows only one single activity and is spanning over 10 to 25 seconds. The dataset provides multilingual human-annotated sentences. With only captions in English considered, there are $85,550$ captions, about 40 captions for each clip. For benchmarking, we follow the common data split of $1,200/100/670$ samples for training/validation/testing [62, 52, 1].

MSR-video to text (MSR-VTT). [58] This dataset contains 10K web video clips and 200K clip-sentence pairs in total. It covers a wide variety of content, and the clips are roughly grouped into 20 categories. Following the instructions on the official website[2] and the settings in [58], the dataset is split into a training set composed of $6,513$ clips, a validation set of 497 clips and a testing set of $2,990$ clips. Each clip is described by 20 single sentences annotated by $1,327$ Amazon Mechanical Turk (AMT) workers. This is one of the largest datasets providing clip-sentence pairs for the video captioning task.

4.2. Implementation details

4.2.1 Data preprocessing & evaluation

By removing those rare words in training split with a threshold of three, we obtain a vocabulary with size of $4,064$ and $10,536$ for MSVD dataset and MSR-VTT dataset respectively, including four additional tokens, *i.e*. *bos*, *eos*, *pad* and *unk*. We do minimum pre-processing to the annotated captions, *i.e*. convert them into lower case and remove punctuation. We add the *bos* and *eos* at the beginning and end of each caption, respectively, and the words that are not contained in vocabulary are replaced with *unk* token. We fix the length of sentences as 30, where we truncate those overlength sentences and add *pad* token at the end of underlength sentences.

To compare the performance of our model with other approaches, we report results on seven model-free automatic evaluation metrics using the Microsoft COCO server [5],

[2] http://ms-multimedia-challenge.com/2017/dataset

Model	Feature	Detector	B@1	B@2	B@3	B@4	M	R	C	Training
SA [60]	VNet+C3D	✗	82.3	65.7	49.7	36.6	25.9	-	-	XE
M3 [56]	VNet+C3D	✗	73.6	59.3	48.3	38.1	26.6	-	-	XE
MA-LSTM [59]	GNet+C3D+A	✗	-	-	-	36.5	26.5	59.8	41.0	XE
VideoLab [40]	Res152+C3D+A+Ca	✗	-	-	-	39.1	27.7	60.6	44.1	XE
v2t_navigator [23]	C3D+A+Ca	✗	-	-	-	42.6	28.8	61.7	46.7	XE
PickNet [6]	Res152+Ca	✗	-	-	-	41.3	27.7	59.8	44.1	RL
RecNet$_{local}$ [55]	InceptionV4	✗	-	-	-	39.1	26.6	59.3	42.7	XE
OA-BTG [65]	Res200	✓	-	-	-	41.4	28.2	-	46.9	XE
MARN [38]	Res101+C3D+Ca	✗	-	-	-	40.4	28.1	60.7	47.1	XE
GRU-EVE [1]	IRV2+C3D+Labels	✓	-	-	-	38.3	28.4	60.7	48.1	XE
POS-CG [54]	IRV2+I3D+Ca	✗	75.7	63.0	50.4	38.3	26.8	60.1	43.4	XE
POS-CG [54]	IRV2+I3D+Ca	✗	80.0	66.4	52.3	39.6	27.5	61.3	50.8	RL
Baseline	IRV2+C3D+Ca	✗	78.9	64.8	51.9	40.5	27.9	59.9	46.1	XE
Baseline	IRV2+C3D+Ca	✓	79.1	65.1	51.4	39.3	27.0	59.9	47.1	XE
SAAT	IRV2+C3D+Ca	✓	80.2	66.2	52.6	40.5	28.2	60.9	49.1	XE
SAAT	IRV2+C3D+Ca	✓	79.6	65.9	52.1	39.9	27.7	61.2	51.0	RL

Table 1: Performance comparisons with different methods on the test set of MSR-VTT dataset in terms of BLEU@1∼4, METEOR, ROUGE_L and CIDEr scores (%). VNet, GNet, C3D, Res-N, IRV2 and A denote VGG19, GoogLeNet, C3D, N-layer ResNet, InceptionResNet-v2 and Audio features, respectively. Ca and Labels denote (20-) Category information provided by the MSR-VTT dataset and object labels by detectors, respectively. XE and RL are short for cross-entropy and reinforcement learning training strategies, respectively.

i.e. BLUE@1∼4) [36] that are precision-based, METEOR [10] that calculates sentence-level similarity scores, CIDEr [51] that is consensus-based, and ROUGE_L [29] that uses longest common subsequence to estimate the similarity between sentences. They are denoted as B@N, M, C, R respectively, where N ranges from 1 to 4. Among them, CIDEr is specially designed for captioning and is considered more consistent with human evaluation [51].

4.2.2 Experimental setup

To increase efficiency as done in [62, 28], we select 28 uniformly-spaced frames from each video clip. We use InceptionResnetV2(IRV2) [46] and C3D [48] as the 2D CNN and 3D CNN, respectively, for feature extraction features. The last avg-pooling layer of the former and the fc6 layer of the latter are considered as the extraction layers. The 2D CNN is pre-trained on ImageNet dataset [44], and Sports 1M dataset [24] is used for the pre-training of C3D. We resize the frames of each video to match the input dimensions of these networks. For the 3D CNN, we use 16-frame clips as inputs with an 8-frame overlap, as done in [1]. Faster R-CNN [42] is used as the object detector in all our experiments. We apply one-hot encoding to each word, and embed them into a 512-dim space. In each iteration, our model loads the features of a mini-batch of 8 video clips on MSVD dataset, and 64 on MSR-VTT dataset. In order to reduce the influence of annotated descriptions that can not be normally parsed, we pass the CIDEr score of each description as the weight to the cross-entropy loss. The Adam [25] optimizer is used for training in our experiments, with a fixed learning rate of 1×10^{-4}. The final performance is determined by the trained model that performs best on the validation set. We use beam search [45] with a beam size of 5 for evaluation.

4.3. Experimental results

Results on MSR-VTT dataset. We comprehensively compare our method against the current state-of-the-arts in video captioning on MSR-VTT dataset. Specifically, we choose i) fundamental methods including SA [60], M3 [56], MA-LSTM [59], VideoLab [40], v2t_navigator [23], ii) latest state-of-the-art methods including PickNet [6], RecNet$_{local}$ [55], OA-BTG [65], MARN [38], GRU-EVE [1], and POS-CG[3]. The Baseline models are implemented by removing the SAAT module for comparison, where the one with detector simulates BUTD [2].

In Table 1 we show the results of different methods on the test set of MSR-VTT dataset. The proposed SAAT model achieves the best performance in terms of CIDEr, BLEU@2 and BLEU@3 while ranking second and third on ROUGH_L and METEOR respectively, when trained by cross-entropy strategy. Using reinforcement learning (SCST [43]), our model achieves the best result in terms of CIDEr and ranks second on the rest metrics. From the comparison, we can see that the methods that fuse multi-modal features show improved results compared

[3]This is reproduced by the released code on `https://github.com/vsislab/Controllable_XGating`

Model	Feature	B@4	M	R	C
S2VT [52]	V+OF	-	29.2	-	-
h-RNN [63]	V+C	49.9	32.6	-	65.8
HRNE [33]	G	43.8	33.1	-	-
LSTM-E [34]	V+C	45.3	31.0	-	-
SCN-LSTM [15]	R152+C	51.1	33.5	-	77.7
DMRM [61]	G+V	51.1	33.6	-	74.8
LSTM-TSA [35]	V+C	52.8	33.5	-	74.0
BAE [3]	R50+C	42.5	32.4	-	63.5
PickNet [6]	R152	46.1	33.1	69.2	76.0
M3 [56]	V+C	52.8	33.3	-	-
MARN [38]	R101+C	48.6	35.1	71.9	92.2
GRU-EVE [1]	IRV2+C	47.9	35.0	71.5	78.1
Baseline	IRV2+C	44.8	33.6	69.0	78.9
SAAT	IRV2+C	46.5	33.5	69.4	81.0

Table 2: Performance comparisons with different methods on the test set of MSVD dataset in terms of BLEU@4, METEOR, ROUGE_L and CIDEr scores (%). V, G and C are short for features from VGGNet19, GoogLeNet and C3D, respectively. OF denotes the optic flow feature.

with SA [60], which indicates the importance of multiple sources of features. High-level semantic information such as Labels used by GRU-EVE [1] contributes to the quality of descriptions, indicated by high CIDEr score. Careful design of the encoder-decoder architecture also benefits the captioning results, as proved by the latest state-of-the-art methods. Compared with these methods and the Baseline, our model explicitly targets the action in videos and greatly improves the CIDEr score.

Results on MSVD dataset. On MSVD dataset, we compare our model trained using cross-entropy strategy against the current state-of-the-art methods in video captioning that strictly follow the train/val/test splits provided by [52], including the Baseline, S2VT [52], hRNN [63], HRNE [33], LSTM-E [34], SCN-LSTM [15], DMRM [61], LSTM-TSA [35], BAE [3], PickNet [6], M3 [56] and GRU-EVE [1].

Table 2 lists the results of different methods. MARN [38] achieves the highest CIDEr score, which indicates that it is rather effective for the small-scale dataset. According to statistics, there are only $882/88/522$ training/validation/testing video clips in this dataset. Compared to its relative performance on the MSR-VTT dataset, it can be inferred that the scene memory method has worse adaptability than ours to new scenes. Except for the MARN, our model outperforms the other methods by a large margin in terms of the CIDEr score.

4.4. Discussion

In this subsection, we perform a quantitative and qualitative evaluation to investigate the effect of our syntax-aware action targeting module. To this end, we first conduct ablation experiments to show the captioning result with different guidance from the module. We used automatic metrics for comparison. Then we evaluate the verb accuracy of different models under different scenes. Finally, we provide multiple examples from the two datasets to show the improvement in the semantic quality of generated captions.

4.4.1 Ablation studies

Different syntax guidance. Actually, our Syntax-Aware Action Targeting module can be seen as a plug-in part that can be easily inserted into existing popular decoders. But here we are more interested in the influence of the specific guidance from this module on the caption decoder, *i.e*. without any guidance (*i.e*. the Baseline), guidance from all the three syntax components (*i.e*. Trip-G), guidance from only *predicate* (*i.e*. SAAT).

Model	B@4	M	R	C	Acc
Baseline	40.5	27.9	59.9	46.1	59.0
Trip-G	39.9	27.2	60.4	46.1	60.5
SAAT	40.5	28.2	60.9	49.1	60.4

Table 3: Performance comparisons of the variants on the test set of MSR-VTT dataset in terms of BLEU@4, METEOR, ROUGE_L, CIDEr scores (%) and the accurate prediction of *predicate* (%).

In Table 3, we show results of the Baseline, Trip-G and SAAT. It can be seen that guidance from syntax components improves captioning results in terms of the ROUGE and CIDEr score. Interestingly, Trip-G model achieves a lower CIDEr score than SAAT. We consider that this is because for some *predicate* such as *running* and *swimming*, the predicted *object* can be *eos* token (like the ground truth generated by NLP tools), which leads to early stopping of the generated captions.

Model	acc-dec	dist-dec	acc-saat	sports	food	cooking
BUTD	55.5	23.4	-	56.5	49.2	63.6
POS-CG	56.6	35.2	-	63.0	46.5	55.5
Baseline	59.0	23.9	-	63.6	50.8	63.6
Trip-G	60.5	17.3	55.3	61.4	55.1	65.5
SAAT	60.4	18.0	56.6	64.5	59.5	69.1

Table 4: Verb accuracy (%) of decoders (acc-dec) and under different scenes (*e.g*. sports), and the average distance to GT-verbs (dist-dec), and verb accuracy (%) of the SAAT module(acc-saat).

Verb accuracy. Given captions generated, we collect both intermediate and decoder's verb accuracy (as well as that of the decoder under different scenes), and the average distance to GT-verbs using official GloVe [39]. Table 4 suggests (1) models with the SAAT module achieve lower dis-

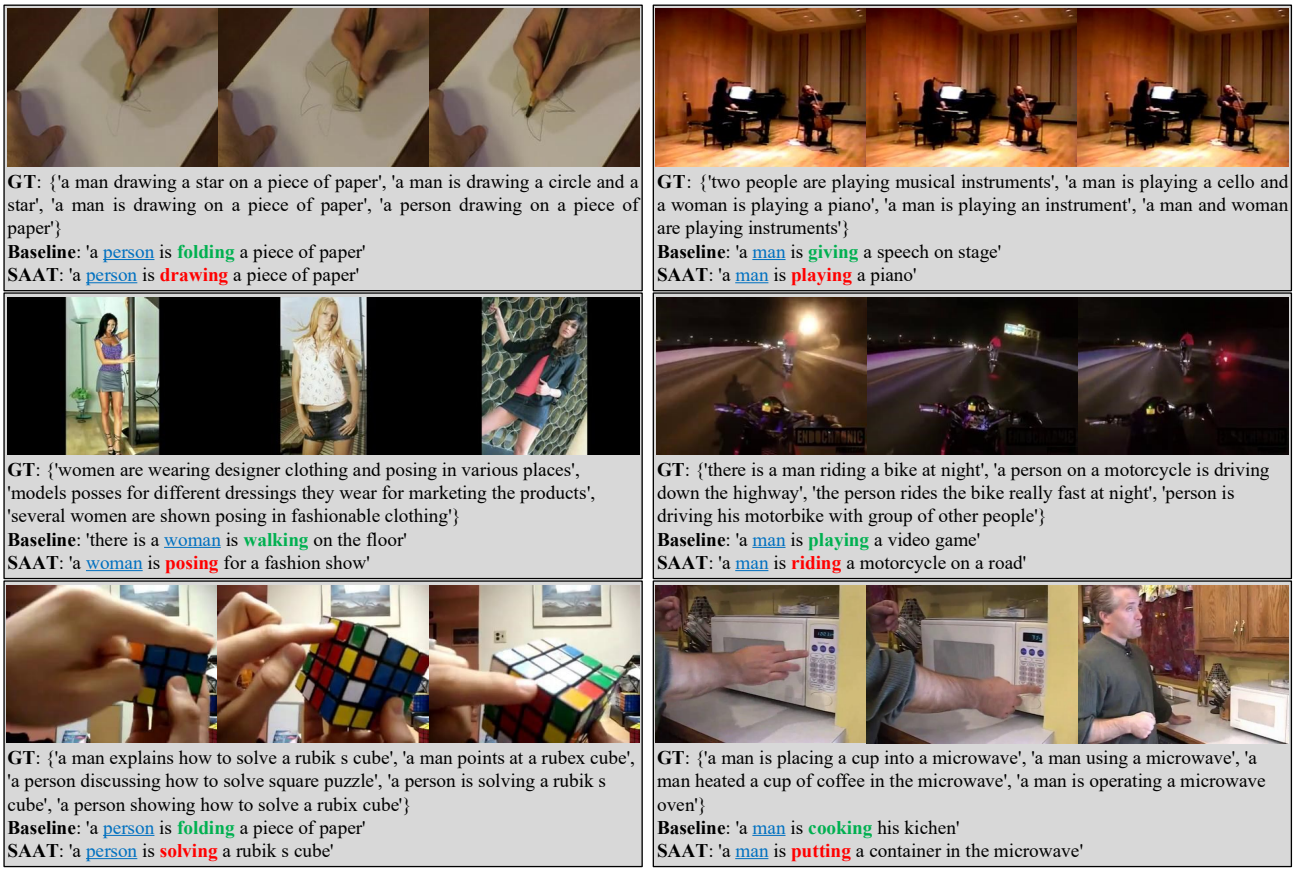

Figure 4: Qualitative comparison between the Baseline and our SAAT model by examples from the test set of MSR-VTT and MSVD datasets. Three frames are shown for each video clip. 3∼5 human annotated descriptions are listed for illustration. Text in blue highlights the *subject* in a sentence. Words in **green** and **red** show the predicted action by Baseline and by SAAT, respectively.

tance and higher verb accuracy in generated captions, especially for the sports/food/cooking scenes that involve finer and diverse actions; and (2) accuracy of the SAAT module is lower than that of the decoder. This is reasonable because the module is designed to provide a coarse direction of verbs, and the decoder learns to predict finer ones.

4.4.2 Qualitative analysis

To provide more insight into what the SAAT module has learned from the video and how it connects vision and language, we present several examples to qualitatively compare our model with the baseline in Figure 4. According to the generated descriptions, we can see that both the Baseline and our SAAT model can correctly predict *subject*, but the former fails to capture the *action* of the video. Due to the limited space, we did not list all the GT descriptions in the figure. The results demonstrate the efficacy of the syntax-aware action targeting module. The results also indicate that improved *action* identification benefits the generated captions, *e.g.* when *drawing*, *posing* are predicted, related scenes such as *a piece of paper*, *a fashion show* are easier to be correctly predicted.

5. Conclusion

In this paper, we proposed a Syntax-Aware Action Targeting (SAAT) model for video captioning that promoted the quality of generated captions. This is achieved by explicitly predicting actions to afford a captioner extra guidance apart from linguistic prior. Though an obvious limitation we observed is that the global temporal information provided by 3D CNNs is not always enough to learn finer actions in video clips, such as distinguishing *cooking* and *eating*, *pushing* and *lifting*. Therefore, we hope that better visual dynamics can be captured to boost the identification of actions, and hence further improve the quality of generated captions.

Acknowledgment

This work was supported by Australian Research Council Projects FL-170100117 and DP-180103424. We would like to thank Jiaxian Guo for initial discussion.

References

[1] Nayyer Aafaq, Naveed Akhtar, Wei Liu, Syed Zulqarnain Gilani, and Ajmal Mian. Spatio-temporal dynamics and semantic attribute enriched visual encoding for video captioning. In *CVPR*, pages 12487–12496, 2019.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018.

[3] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Hierarchical boundary-aware neural encoder for video captioning. In *CVPR*, pages 1657–1666, 2017.

[4] Andrei Barbu, Alexander Bridge, Zachary Burchill, Dan Coroian, Sven Dickinson, Sanja Fidler, Aaron Michaux, Sam Mussman, Siddharth Narayanaswamy, Dhaval Salvi, et al. Video in sentences out. In *UAI*, page 102–112, 2012.

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[6] Yangyu Chen, Shuhui Wang, Weigang Zhang, and Qingming Huang. Less is more: Picking informative frames for video captioning. In *ECCV*, pages 358–373, 2018.

[7] Leonard Dahlmann, Evgeny Matusov, Pavel Petrushkov, and Shahram Khadivi. Neural machine translation leveraging phrase-based models in a hybrid search. In *EMNLP*, 2017.

[8] Bo Dai and Dahua Lin. Contrastive learning for image captioning. In *NeurIPS*, pages 898–907, 2017.

[9] Pradipto Das, Chenliang Xu, Richard F Doell, and Jason J Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *CVPR*, pages 2634–2641, 2013.

[10] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *SMT*, pages 376–380, 2014.

[11] Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G Schwing, and David Forsyth. Fast, diverse and accurate image captioning guided by part-of-speech. In *CVPR*, pages 10695–10704, 2019.

[12] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634, 2015.

[13] Jianfeng Dong, Xirong Li, Weiyu Lan, Yujia Huo, and Cees GM Snoek. Early embedding and late reranking for video captioning. In *ACM Multimedia*, pages 1082–1086, 2016.

[14] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *CVPR*, pages 9346–9355, 2019.

[15] Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. Semantic compositional networks for visual captioning. In *CVPR*, pages 5630–5639, 2017.

[16] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *CVPR*, pages 4089–4098, 2018.

[17] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, pages 2712–2719, 2013.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[19] Xinwei He, Baoguang Shi, Xiang Bai, Gui-Song Xia, Zhaoxiang Zhang, and Weisheng Dong. Image caption generation with part of speech guidance. *Pattern Recognition Letters*, 2017.

[20] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. In *NeurIPS*, pages 11135–11145, 2019.

[21] Po-Sen Huang, Chong Wang, Sitao Huang, Dengyong Zhou, and Li Deng. Towards neural phrase-based machine translation. In *ICLR*, 2018.

[22] Mohit Iyyer, Jordan Boyd-Graber, and Hal Daumé III. Generating sentences from semantic vector space representations. In *NeurIPS workshop on learning semantics*, 2014.

[23] Qin Jin, Jia Chen, Shizhe Chen, Yifan Xiong, and Alexander Hauptmann. Describing videos using multi-modal fusion. In *ACM Multimedia*, pages 1087–1091, 2016.

[24] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[26] Atsuhiro Kojima, Takeshi Tamura, and Kunio Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 50(2):171–184, 2002.

[27] Rémi Lebret, Pedro O Pinheiro, and Ronan Collobert. Phrase-based image captioning. In *ICML*, 2015.

[28] Xuelong Li, Bin Zhao, and Xiaoqiang Lu. Mam-rnn: Multi-level attention model based rnn for video captioning. In *IJCAI*, pages 2208–2214, 2017.

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

[30] Huan Ling and Sanja Fidler. Teaching machines to describe images via natural language feedback. In *NeurIPS*, pages 5068–5078, 2017.

[31] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, pages 375–383, 2017.

[32] Anh Nguyen, Dimitrios Kanoulas, Luca Muratore, Darwin G Caldwell, and Nikos G Tsagarakis. Translating videos to commands for robotic manipulation with deep recurrent neural networks. In *ICRA*, pages 1–9, 2018.

[33] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video rep-

resentation with application to captioning. In *CVPR*, pages 1029–1038, 2016.

[34] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, pages 4594–4602, 2016.

[35] Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. Video captioning with transferred semantic attributes. In *CVPR*, pages 6504–6512, 2017.

[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002.

[37] Ramakanth Pasunuru and Mohit Bansal. Reinforced video captioning with entailment rewards. In *EMNLP*, pages 979–985, 2017.

[38] Wenjie Pei, Jiyuan Zhang, Xiangrong Wang, Lei Ke, Xiaoyong Shen, and Yu-Wing Tai. Memory-attended recurrent network for video captioning. In *CVPR*, pages 8347–8356, 2019.

[39] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[40] Vasili Ramanishka, Abir Das, Dong Huk Park, Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, and Kate Saenko. Multimodal video description. In *ACM Multimedia*, pages 1092–1096, 2016.

[41] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[43] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, pages 7008–7024, 2017.

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[45] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NeurIPS*, pages 3104–3112, 2014.

[46] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017.

[47] Ying Hua Tan and Chee Seng Chan. Phrase-based image captioning with hierarchical lstm model. In *ACCV*, 2017.

[48] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[49] Kiyotaka Uchimoto, Hitoshi Isahara, and Satoshi Sekine. Text generation from keywords. In *ACL*, pages 1–7, 2002.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[51] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015.

[52] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *ICCV*, pages 4534–4542, 2015.

[53] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL*, pages 1494–1504, 2015.

[54] Bairui Wang, Lin Ma, Wei Zhang, Wenhao Jiang, Jingwen Wang, and Wei Liu. Controllable video captioning with pos sequence guidance based on gated fusion network. In *ICCV*, 2019.

[55] Bairui Wang, Lin Ma, Wei Zhang, and Wei Liu. Reconstruction network for video captioning. In *CVPR*, pages 7622–7631, 2018.

[56] Junbo Wang, Wei Wang, Yan Huang, Liang Wang, and Tieniu Tan. M3: Multimodal memory modelling for video captioning. In *CVPR*, pages 7512–7520, 2018.

[57] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Visual question answering: A survey of methods and datasets. *CVIU*, 163:21–40, 2017.

[58] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296, 2016.

[59] Jun Xu, Ting Yao, Yongdong Zhang, and Tao Mei. Learning multimodal attention lstm networks for video captioning. In *ACM Multimedia*, pages 537–545, 2017.

[60] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015.

[61] Ziwei Yang, Yahong Han, and Zheng Wang. Catching the temporal regions-of-interest for video captioning. In *ACM Multimedia*, pages 146–153, 2017.

[62] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher Pal, Hugo Larochelle, and Aaron Courville. Describing videos by exploiting temporal structure. In *ICCV*, pages 4507–4515, 2015.

[63] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, pages 4584–4593, 2016.

[64] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, pages 3165–3173, 2017.

[65] Junchao Zhang and Yuxin Peng. Object-aware aggregation with bidirectional temporal graph for video captioning. In *CVPR*, pages 8327–8336, 2019.

[66] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. Grounded video description. In *CVPR*, 2019.