Interpretable Video Captioning via Trajectory Structured Localization

Xian Wu[1] Guanbin Li[1*] Qingxing Cao[1], Qingge Ji[1] Liang Lin[1,2]
[1]Sun Yat-sen University [2]SenseTime Group Limited
sysuwuxian@gmail.com, liguanbin@mail.sysu.edu.cn, caoqx@mail2.sysu.edu.cn,
issjqg@mail.sysu.edu.cn, linliang@ieee.org

Abstract

Automatically describing open-domain videos with natural language are attracting increasing interest in the field of artificial intelligence. Most existing methods simply borrow ideas from image captioning and obtain a compact video representation from an ensemble of global image feature before feeding to an RNN decoder which outputs a sentence of variable length. However, it is not only arduous for the generator to focus on specific salient objects at different time given the global video representation, it is more formidable to capture the fine-grained motion information and the relation between moving instances for more subtle linguistic descriptions. In this paper, we propose a Trajectory Structured Attentional Encoder-Decoder (TSA-ED) neural network framework for more elaborate video captioning which works by integrating local spatial-temporal representation at trajectory level through structured attention mechanism. Our proposed method is based on a LSTM-based encoder-decoder framework, which incorporates an attention modeling scheme to adaptively learn the correlation between sentence structure and the moving objects in videos, and consequently generates more accurate and meticulous statement description in the decoding stage. Experimental results demonstrate that the feature representation and structured attention mechanism based on the trajectory cluster can efficiently obtain the local motion information in the video to help generate a more fine-grained video description, and achieve the state-of-the-art performance on the well-known Charades and MSVD datasets.

1. Introduction

Video captioning which aims at automatically describing videos containing rich and open-domain activities with natural language sentences, is a core problem towards high-level video understanding and has recently received increasing interest in both computer vision and artificial intelligence communities. It has a variety of practical applications including human-robot interaction, video indexing, and describing movies for the blind. Despite recent progress, video captioning remains a very challenging problem that calls for more accurate solution.

Conventional video captioning algorithms are based on template-based methods [8, 17, 29], which works by predefining a serious of sentence generation template with some specific grammar rules, and adaptively correlating each part of the sentence with detected object, object properties as well as object relationship from video content analysis. Though this kind of method is simple and intuitive, they suffer from limited sentence templates and can only generate very rigid sentence descriptions. Benefit from the rapid development of deep neural networks, especially Recurrent Neural Network (RNN), sequence learning method have recently been widely used in video captioning and achieved very inspiring results. It has quickly become the mainstream framework for solving the problem [21, 30]. This kind of method is primarily based on an encoder and decoder mechanism to recurrently map the feature embedding of a video clip to a word sequence. Specifically, an encoder neural network (CNN or RNN) reads the video frames and generate a compact video representation, which is in turn fed to a decoder RNN or its variants (LSTM, GRU, etc) to generate a natural sentence word by word. RNN based sequence learning method is originally inspired by the recent advances in neural machine translation and has attracted a line of improved work ever since its emergence.

However, all these methods utilize global features extracted at image-level for video feature representation, ignoring the movement details of various objects within the video, they thus can only generate very general descrip-

[*]Corresponding author is Guanbin Li. This work was supported by the National Natural Science Foundation of China under Grant 61702565 and Grant 61671182, the Science and Technology Planning Project of Guangdong Province under Grant 2017B010116001 and the Nature Science Foundation of Guangdong Province under Grant 2016A030313288. This work was also sponsored by CCF-Tencent Open Research Fund.

tion (e.g. “cooking in the kitchen” vs “slice the tomatoes and put them in the pot”). Though temporal structure has been modeled, they usually can only accurately describe uncomplicated video activities with a single dominant object. On the other hand, attention mechanism can be applied to establish an explicit relationship between the generated sentence and the content in the video and thus provides an interpretable mechanism for the generated sentence description. However, existing attention modeling used in video captioning only confines the attended target to one entire frame or some specific object in a single frame image, ignoring the influence of local motion information on the refined description. In fact, the motion of objects and their parts are very informative cue for video comprehension. Imagine we see a person holding a plate, we have to watch a sequence of motion trajectory of the hand and the plate before we can conclude that he is to wash dishes or to take the dishes.

Moreover, the previous method treats each sentence as a chain structure, ignoring the semantic structure of the sentence. All identified objects are fed into the attention model in spite of their importance and relations in video, which are indeed less discriminative for major content of video. We argue that structure information exists in the sentence is crucial for video caption. For instance, we watch a video regarding a man siting on a chair playing with a phone, it is only when we have captured the overall configuration of the man and the chair we can conclude that he is using a phone rather than talking to a phone. However, how to encode the sentence structured information into video caption remains a challenge as we lack groundtruth sentence and are thus not able to get predefined semantic structured information during inference.

To address the two problems above, we propose a trajectory structured attentional encoder-decoder framework (TSA-ED) which works by incorporating an attentive structured localization mechanism in a prevailing LSTM-based encoder and decoder framework. In particular, our proposed TSA-ED is composed of a pre-processing stage for trajectory cluster feature representation and a structured aware encoder-decoder network framework. In the pre-processing stage, we extract a set of trajectory cluster features. Each trajectory cluster well captures one specific local motion pattern and it is used in the decoding phase for local spatial-temporal feature attention. During the decoding phase, we dynamically change the feature vectors of candidate spatial-temporal regions in video and simultaneously generate the caption. Specifically, based on the structure parsed from the sentence and the corresponding mapping between word and related motion region, we can extract the phrase-level corresponding spatial-temporal feature to model the overall configuration of motion objects which serves as a candidate feature vector for subsequent word prediction.

In summary, this paper has the following contributions:

- We propose a novel trajectory structured attentional network which fully consider both the motion information and the sentence semantic structure with an attentive structured localization mechanism. It is able to generate more elaborate and more accurate video captioning than existing traditional global image feature or static object representation based methods.
- The attentive trajectory localization mechanism can be regarded as an effective visualization tool and can greatly enhance the model’s interpretability, so that we can roughly obtain the motion sequence corresponding to each word or clause while performing sentence inference.
- The proposed TSA-ED method achieves the state-of-the-art performance on the Charades and MSVD datasets with different evaluation metrics in our experiments.

2. Related work

Previous video captioning algorithms are based on template-based methods, which are mainly based on a two-step approach, including role-word detection(eg.,subject, verb and object) and language grammar rules definition. In such works, the sentence for video description is first split into parts, each of which is aligned with visual content. For example, [17] learns a Conditional Random Field (CRF) to model the relationships between different components of the input video and generate description for video. However these methods are insufficient to model the richness of visual and semantic information in video captioning. Recently, benefit from the rapid development of deep learning, video captioning has made great success and lots of research works have proposed to use recurrent neural networks to generate video descriptions. The baseline of encoder-decoder framework was first proposed in [22], which used CNN-based mean pooling method to encode the video frame-level information and adopt a RNN to decoder the sentence. More recently, inspired by attention mechanism which has made great success in image captioning, multi-label classification [27, 4], and natural language processing, many works [30, 31] use soft attention method to selectively attend most salient video frame to generate video description. However, existing attention modeling used in video captioning only confines the attended target to one entire frame or some specific object in a single frame image, ignoring the influence of local motion information on the refined description.

Dense Trajectory [23] and its improved version: improved Dense Trajectories [24] have made great success

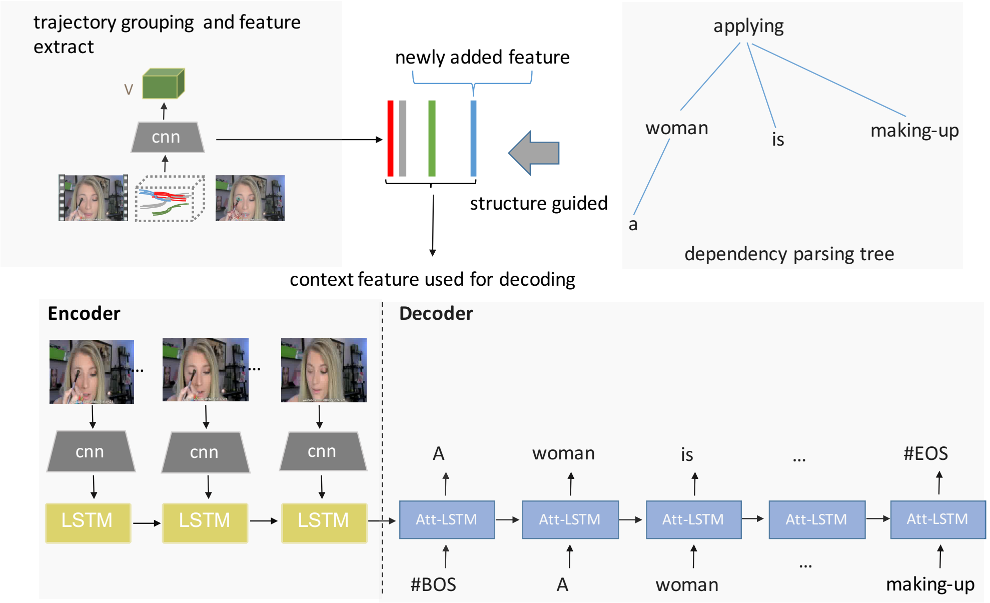


Figure 1. The overview of the Trajectory Structured Attentional Encoder-Decoder Network framework. Our method consists of an encoding phrase for global feature representation and an decoding stage for generating video description by incorporating motion information and the sentence semantic structure.

in video recognition and general video classification tasks. Beside the hand-crafted visual features like Dense trajectories, researchers have started exploring CNN on video representation. [19] proposed two stream networks to combine optical flow frame and RGB stream as inputs to train the ConvNets, which achieved comparable performance with the state-of-the-art hand-crafted features. More recently, [25] proposed trajectory-pooled deep convolutional descriptor (TDD), which combines the advantages of hand-crafted features and deep learning feature. In this paper, we argue that the trajectory-based motion information can also be used to improve the video caption results, and thus utilize the trajectory motion information in the detail enhancement of video descriptions generation.

Recently, some work explore to incorporate sentence parsing to solve visual tasks. For example, [3] utilizes the parsed image entities' relation to enhance the caption results. [11] proposes to use parsed sentence to learn phrase level corresponding between image region and text phrase. However, Our work can be regarded as the first effort to leverage the structure of sentence in video captioning. Different from most of the previous sequence attention model which attend video frame or frame region, our work contributes to fully exploit the video temporal information by attending the trajectory clusters and the relation between motion instances and thus provides the explanation for video captioning.

3. Trajectory Structured Attentional Encoder-Decoder Network

In this section, we devise our customized encoder and decoder architecture with incorporated trajectory structured attention mechanism. As shown in Fig. 1, our proposed method framework consists of a pre-processing stage for trajectory feature extraction, an encoder phase for global feature representation and a decoder stage for generating video description sentence by structured attending corresponding motion information for the predicted word.

3.1. Trajectory Grouping and Feature Extraction

Video trajectory feature has been widely recognized as a superior description for motion information, and has demonstrated very promising results in video classification as well as activity recognition. In this paper, we resort to point trajectory feature for more refined video description generation. We first refer to [16] and extract dense point trajectories $T = \{T_1, T_2, ..., T_N\}$ over the entire video volume, where N is the number of trajectories and T_i denotes the i^{th} trajectory in the video. These trajectories are calculated by first densely sampling a set of points on a grid and performing sample points tracking by media filtering of dense flow field. To avoid the drifting problem of tracking, the maximum length of trajectory is set as 15-frame. To group the trajectories, we compute the similarity between each trajectory pair and form an $N \times N$ affinity matrix. We use the distance metric proposed in [16] for grouping, which considers the temporal overlapping, spatial proximity and speed similarity in evaluation. We also threshold on the trajectory pair distance and enforce the affinity to be zero for those are not spatially close (i.e., distance larger than a default threshold). Following [16], we further apply graph based clustering method and partition the detected trajectories into groups based on the affinity matrix, each group is called a trajectory cluster. Given a video V, we can finally obtain m trajectory clusters.

Given the i^{th} trajectory cluster $\mathrm{TC}(i) = \{T_{i1}, T_{i2}, ..., T_{iK}\}$ with K trajectories, we denote each trajectory as a position point sequence $T_{ik} = \{(x_{ik}^1, y_{ik}^1, z_{ik}^1), (x_{ik}^2, y_{ik}^2, z_{ik}^2), ..., (x_{ik}^L, y_{ik}^L, z_{ik}^L)\}$, with $(x_{ik}^l, y_{ik}^l, z_{ik}^l)$ being the 3D coordinates of the l^{th} point in trajectory T_{ik} and L being the length of the trajectory. Following [25], we apply deep convolutional network and describe each trajectory as a trajectory-pooled deep-convolutional descriptor (TDD). In principle, any kind of ConvNet architecture can be selected for TDD extraction. Without loss of generality, assume that we extract feature from one selected output feature map of a ConvNet. We first separately feed each frame image to the ConvNet to generate a feature map of size $H \times W \times N$, where H, W and N are respectively the height, width and number of channels of the output feature map, concatenating all the feature maps along the duration of the video, we can finally obtain an entire feature map $C \in R^{H \times W \times L \times N}$, with L being the length of the video clip. Given this video feature map, a trajectory point with coordinates $(x_p, y_p, z_p$ will be center on $(r \times x_p$, $r \times y_p$, $r \times z_p)$ in feature map, where r denotes the map size ration with respective to the input size. The feature of T_{ik} can thus be calculated as

$$F_{ik} = \sum_{p=1}^{L} C(r \times x_{ik}^p, r \times y_{ik}^p, r \times z_{ik}^p)/L, \qquad (1)$$

and the feature vector of the trajectory cluster is computed as the mean pooling of all included trajectory feature vectors, Denoted as $F_i = \sum_{k=1}^{K} F_{ik}/K$.

3.2. Encoder-Decoder Framework

The encoder-decoder framework is composed of two networks, including the encoder and the decoder. The encoder network EN learns to encode the input data x into a sequence of feature vectors: $V = \{v_1, v_2, ..., v_k\} = \mathrm{EN}(x)$. The architecture choice of the EN depends on the type of input data. For example, for static spatial data (e.g. image), it is natural to choose convolutional neural network, for temporal and sequential data, recurrent neural network (RNN) or its variants are very good options, while for spatial-temporal data, such as videos, a combination of CNN and RNN is a good alternative. The decoder network takes the encoder representation V as input and learns to generate the output y. As with the encoder, the architecture of the decoder can be CNN or RNN, depending on the type of y. For video captioning, as the output is a word sequence, RNN is a method of choice. The RNN decoder DN runs sequentially to produce the output sequence $y = \{y_1, y_2, ..., y_n\}$, with element y_t denoting the predicted word at t^{th} time step. Specifically, at each time step t, the RNN updates its hidden state h_t based on its previous stage h_{t-1}, the previous output y_{t-1} as well as the encoder embedding V, and calculates the output y_t as:

$$\begin{bmatrix} y_t \\ h_t \end{bmatrix} = \mathrm{DN}\left(h_{t-1}, y_{t-1}, V\right) \qquad (2)$$

3.3. Encoder: LSTM Encoding on Temporal Segments

Deep convolutional neural networks have recently achieved many successes in visual recognition tasks and the pre-trained CNN models for object classification have been demonstrated very effective feature extractor for other vision tasks such as object detection and visual captioning. In this paper, we resort to the most successful classification model called deep ResNet [9] for frame image feature extraction. Inspired by the temporal segment networks proposed in [26], we exploit a sequence of short snippets sparsely sampled from the entire video for approximate video feature representation, which has been proved an efficient video feature representation for accurate action recognition. Given a video clip, we first divide it into S segments of equal time interval and randomly sample a frame from each of them. Then we separately feed each sampled frame image to a pre-trained ResNet model and extract the deep feature vector from the pool5 layer with 2048-Dimension. Let $\mathrm{FG} = \{F_1, F_2, ..., F_S\}$ denote the extracted global feature of the video clip. We further add one linear transform layer to transform the feature vector of each frame into new feature embedding, which aims at dimensionality reduction. Finally, we can obtain S deep feature embedding, denoted as $V = \{V_1, V_2, ..., V_S\}$. To better model the sequential and temporal information of the video, we apply LSTM network to encode V into a global feature representation with fixed length. The LSTM network consists of S steps, and learns to predict a feature embedding at each step s by looking at the previous LSTM hidden state h_{t-1} and the feature input V_s of current stage, the output is formulated as $h_t = \mathrm{LSTM}(h_{t-1}, V_t)$. Finally, the output feature vector of the last hidden stage h_S is treated as the global video feature representation, which serves as initial input for caption generation in the decoder network.

3.4. Decoder: Structured Attention LSTM for Video Description Generation

Inspired by the good performance of applying RNN in visual caption generation, we propose to use LSTM for video description generation, but incorporate with tree structured semantic attention. As illustrated in Fig 1, we devise our trajectory structured LSTM video captioning model by injecting both the global video representation produced by the encoder and the attentive tree structured motion feature into LSTM.

Before we introduce our method, we first review the transition-based dependency parsing [2] and then show how to incorporate semantic relation parsed from the tree into video caption framework. Here we employ the arc-standard system [2] as the basis of parser. In the arc-standard system, a configuration $c = (S, B, A)$ consists of a stack S, a buffer B, and a set of dependency arcs A. The initial configuration for a sentence $\{w1, ..., w_n\}$ is S = [$root$], B = [$w_1, ..., w_n$], A = $\emptyset$. A configuration c is terminal if the buffer is empty and the stack contains the single node $root$, and the parse tree is given by A_c. Denoting $s_i(i = 1, 2, ...)$ as the i_{th} top element on the stack, and b_i as the i^{th} element on the buffer, the arc-standard system define three types of transitions:

- Left-arc(l): adds an arc $s_1 \rightarrow s_2$ with label l and removes s_2 from the stack.
- Right-arc(l): adds an arc $s_2 \rightarrow s_1$ with label l and removes s_1 from the stack.
- Shift: moves b_1 from the buffer to the stack.

a dependency parse tree will be given by arcs A as we continue to execute this transition sequence until the c is terminal.

Given a video V and its corresponding sentence $\{w_1, w_2, ..., w_n\}$, we extract p trajectory clusters from the video serving as the basis of the feature vectors F, the feature of which are denoted as $\{F_1, F_2, ..., F_p\}$. We also get the transition sequence of the sentence, which are denoted as $\{T_1, T_2, ..., T_k\}$. At the beginning, the stack S = [$root$] and the buffer B = [$w_1, w_2, ..., w_n$]. we repeated perform the state transition until the configuration is terminal.

- If current T is equal to shift, a word w_i will be moved from the buffer to the stack. The process is similar to generating next word in visual caption if we consider the current stack state as a sentence is being generated, so we utilize the soft attention based method to predict next word w_i from the current feature vectors F. Specifically, given the feature matrix of m feature vectors $V_{TC} \in \Re^{d*m}$ with each row corresponds to a d dimension feature vector and the hidden state $h_{t-1} \in \Re^d$ of the LSTM, we feed them to a single linear transform layer followed by a softmax function to calculate the attention distribution over the m feature vectors of a given video:

$$z_t = w_h^T \tanh\left(W_v V_{TC} + (W_g h_{t-1})\, \mathbf{1}^T\right) \quad (3)$$

$$\alpha_t = \mathrm{softmax}\,(z_t) \quad (4)$$

where $W_v, W_g \in \Re^{m*d}$ and $w_h \in \Re^m$ are the parameters to be learned. $\mathbf{1} \in \Re^m$ is a vector with all elements set to 1. $\alpha_t \in \Re^m$ is the attention weight of m feature vectors. Given the attention distribution, our context feature vector is calculated as:

$$c_t = \sum_{i=1}^{m} \alpha_{ti} F_i. \quad (5)$$

The attention distribution allows the decoder to selectively focus on some salient motion groups at different time and simultaneously consider their context information, then the current hidden state h_t is updated as:

$$h_t = \mathrm{LSTM}(h_{t-1}, c_t, w_t) \quad (6)$$

The probability over a vocabulary of possible words at time t can be calculated as:

$$p_t = \mathrm{softmax}(W_p h_t + b), \quad (7)$$

with W_p, b being the parameters to be learned in the network. Suppose maximum attention weight among α_t is α_{tj}, then we mapping w_i to F_j to indicate that w_i is most related to the visual region F_j.

- If current T is equal to left-arc, according to the transition system, we adds an arc $w_i \rightarrow w_j$ and remove w_j from the stack, then we dynamically change the feature vectors of F as following: suppose w_i and its child nodes $\{w_i^1, w_i^2, ..., w_i^t\}$ in dependency tree mapping features are $\{F_i, F_i^1, F_i^2, ..., F_i^t\}$, all these $(t+1)$ features will be eliminated from the F, and a new feature vector F_{new} which based on these $(t+1)$ features will be added into the F. To compute the F_{new}, since the position of these $(t+1)$ spatial temporal proposals positions are known, we can choose a minimum spatial temporal proposal which contain all of them and then extract this proposal's feature as our newly feature vector. Specifically, we do roi pooling [7] on convolutional feature maps over time to represent the newly added feature vector F_{new}. We argue that the F_{new} reflects the relation information between moving object which exists under the subtree rooted in S_i.The situation when T is equal to right-arc is similar to the situation T is equal to left-arc.

4. Network Training and Description Generation

The goal of video captioning is to generate a word sequence for a given video. The calculation of each word can be treated as predicting a distribution over a word vocabulary:

$$P(w_t | w_{1:t-1}, V, V_{TC}, W) \quad (8)$$

where w_t denotes the embedding of the t^{th} word and W represents the model parameters. The network model learns to predict a word at each step by looking at the global video feature V, the features of the trajectory clusters V_{TC} and the previous predicted words. The overall loss function for optimization is set to be the log-likelihood over the entire training set.

$$\sum_{i=1}^{N}\sum_{j=1}^{t_i} - \log P(w_j^i | w_{1:j-1}^i, V_i, V_{iTC}, W) \qquad (9)$$

where N is total number of training video-description pairs and t_i denotes the number of words within description w^i. We use Stochastic Gradient Descent to find the optimum with gradient computed via Backpropageation through time dimension[28].

During the testing phase, since we don't have ground truth sentence, we cannot get the corresponding transition sequence. However, during decoding, when a word is predicted, the newly generated word would be added into the stack, so the stack information is complete. Like [2], we train a dependency parser only depended on stack state, for more details, please refer to [2]. By this way, with the learned parser, we can determine the next transition state according the current stack information and generate the sentence like the way we do in training.

We choose BeamSearch method [21] to generate descriptions for a testing video.In our experiment, we set the beam size k to 5.

5. Experiments

We test the proposed approach on two benchmark datasets, including MSVD and Charades. In this section, we introduce our experiments and analyze the results in details.

5.1. Datasets

MSVD. This dataset contains 1970 short video clips collected from YouTube. Each video clip typically describes a single activity in open domain and is annotated with roughly 40 English descriptions. For fair comparison, we follow the splits setting provided in [21], with 1200 videos for training, 100 videos for validation , and 670 videos for testing.

Charades. This dataset is more challenging. It consists of 9848 videos with an average length of 30 seconds. Different from MSVD datasets, it focuses on common household activities in indoors scenes which can be quite diverse in activity and places. Following [18], we split the datasets into three parts, including 7569 for training, 400 for validation and 1863 for testing.

5.2. Evaluation Metrics

We employ three popular metrics for evaluation, including BLEU [15], METEOR [1] and CIDEr [20].

BLEU is the most popular metric for evaluation of machine translation performance and it is based on the n-gram precision. As with previous works, we choose 4-gram for evaluation. The METEOR metric is calculated base on the alignment between a given hypothesis sentence and a set of candidate reference sentences. It works by comparing exact token matches, stemmed tokens, paraphrase matches, as well as semantically similar matches using WordNet synonyms. CIDEr, on the other hand, computes the average cosine similarity between n-grams found in the generated description and those found in reference sentences, weighting them using TF-IDF. The authors of CIDEr [20] reported that CIDEr and METEOR are always more accurate, especially when the number of reference captions is low. To ensure a fair evaluation, we use the Microsoft COCO evaluation toolkit to compute all scores, as done in previous video captioning works([21], [30]).

5.3. Experimental Settings

For video segment frame representation, we first divide the video into 3 segments, randomly sample a frame from each segment and take the output of 2048-dimension from the output of the pool5 layer of ResNet152 [9] as the segment frame representation. While performing the representation of video trajectory clusters, we take the output of last convolutional feature map of ResNet152 [9] for TDD extraction. For sentence representation, We first convert all descriptions to lower case, remove all punctuations and tokenize the sentences. After preprocessing, This yields a vocabulary of 12,593 in size of the MSVD dataset and a vocabulary of 3681 in size for the Charades dataset, Then each word in the sentence is represented as “one hot” vector(binary index vector in a vocabulary). We use Stanford parser[2] to parse the training sentence and get corresponding transition sequence.

During the training phase, in order to deal with sentences with arbitrary length, we add a begin-of-sentence tag BOS to start each sentence and an end-of-sentence tag EOS to end each sentence. In the testing phase, we input the BOS into video decoder to start generating video descriptions. In addition, The dimension of hidden size in LSTM is set to 1000, the word embedding size and frame embedding size are both set 500. Empirically, our objective function Eq. 9 is optimized over the whole training video sentence pairs with mini-batch 30 in size of MSVD and Charades. We adopt adam [10], which is an adaptive learning rate approach, to optimize our target loss function. In addition, we utilize dropout regularization with the rate of 0.5 in all layers and clip gradients element wise at 10. We stop training our model until it reaches 30 epochs or the evaluation metric does not improve on the validation set at the patience of 20. In testing stage, we adopt the beam search strategy and set the beam size to 5. We implement our proposed model on the open source computing framework Torch7 [5].

5.4. Comparison with the State of the Art

Quantitative Analysis. We compare our proposed trajectory structured attentional encoder-decoder model (TSA-ED) against the state-of-the-art methods over the MSVD dataset and the Charades dataset. Table 1 demonstrates the performances of different models on the MSVD dataset. We compare our method with six state-of-the-art methods with different parameters setting, including basic encoder-decoder model (S2VT) [21], soft attention based LSTM network (SA) [30], hierarchical processing based decoders (HRNE) [12], joint learning based LSTM embedding network(LSTM-E) [13], P-RNN [31] and transferred semantic attributes (LSTM-TSA) [14]. As can be seen, our proposed method consistently outperforms all the state-of-the-art techniques in terms of all the three evaluation metrics. In particular, the CIDEr of our TSA-ED model achieves 74.9%, which makes relative improvement over SA, p-RNN and LSTM-TSA by 42.56%, 37.5% and 3.08%, respectively. Similar conclusions can be drawn in the other two metrics. Noted that although we only sample several static frames for global video representation, our proposed method greatly outperforms other encoder-decoder framework based models with much more complex global feature modeling (e.g. VGGNet/GoogleNet and optical flow/C3D for spatial and temporal modeling, attributes from video), which demonstrates the complementary effectiveness of the trajectory structured attention mechanism. For fair comparsion, we also report the performance of our method with VGG and VGG+C3D features on MSVD dataset. As shown in Table 1, though the use of inferior feature leads to a certain degree of performance degradation, it can still outperform its counterpart (with same feature setting) when compared to the models listed in Table 1 of the paper.

The performance comparisons on Charades dataset is summarized in Table 2. The scores of the three evaluation metrics on this dataset are much lower than those on MSVD, due to the much higher complex visual and textual content in this dataset. Our proposed method can also greatly outperform other state-of-the-art methods in this dataset. Specifically, our TSA-ED model makes the relative improvement over the best best-performing existing algorithm (MAAM) by 1.14%, 24.55% and 17.39% respectively on the evaluation metric of METEOR, CIDEr and BLEU_4.

Method	METEOR	CIDEr	BLUE_4
S2VT (V) [21]	29.2	-	-
S2VT(V+O) [21]	29.8	-	-
SA(G+C) [30]	29.6	51.7	41.9
HRNE(G) [12]	32.1	-	43.6
HRNE-SA(G) [12]	33.1	-	43.8
HRNE-SA(G+V) [12]	33.9	-	44.3
LSTM-E(V) [13]	29.5	-	40.2
LSTM-E(C) [13]	29.9	-	41.7
LSTM-E(V+C) [13]	31.0	-	45.3
p-RNN(V) [31]	31.1	62.1	44.3
p-RNN(C) [31]	30.3	53.6	47.4
p-RNN(V+C) [31]	32.6	65.8	49.9
LSTM-TSA(V+C+A) [14]	32.4	71.5	50.2
Ours(V)	32.3	65.6	45.4
Ours(V+C)	33.5	73.5	50.1
Ours(R)	**34.0**	**74.9**	**51.7**

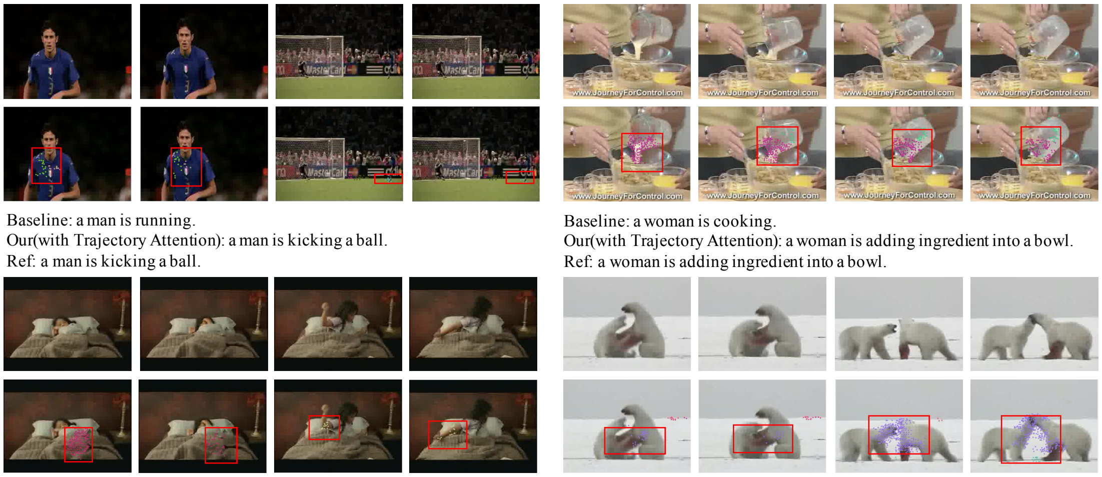
Table 1. Comparsion with state of art methods in terms of METEOR, CIDEr and BLUE_4 over the MSVD test set. − indicates unknown scores. Methods of the same name but different text in the brackets indicates the same method with different feature setting.

Method	METEOR	CIDEr	BLUE_4
S2VT [21]	16.0	14.0	11.0
SA [30]	14.3	18.1	10.8
MAAM [6]	17.6	16.7	11.5
Ours	**17.8**	**20.8**	**13.5**

Table 2. Comparison with state of art methods in terms of METEOR, CIDEr and BLUE_4 over the Charades test set.

Qualitative Analysis. Although the model-free evaluation metrics (i.e. BLEU, METEOR, CIDEr) have demonstrated the superiority of our proposed framework to other methods, it is still not intuitive how those performance gain reflects to the quality of generated video description and how the incorporated trajectory attention mechanism acts on generating refined video captioning. In Figure 2, we present some sample video clips and their corresponding descriptions, both generated and reference. For each sample video, we list the generated description by our proposed method and those generated from the baseline encoder-decoder framework for comparison. As shown in the figure, our proposed TSA-ED can provide more specific, comprehensive and accurate description than baseline encoder-decoder framework, as the incorporated trajectory contains more local appearance feature for inference. For example, on the top-left panel, our proposed model can well capture the “football” and the “kicking” action and correctly generate the description “a man is kicking a ball” instead of roughly saying “a man is running”. Besides, the trajectory attentional encoder-decoder framework is also adept at capturing the fine-grained object motion information for more subtle linguistic descriptions. For example, we can see from the top-right panel of the figure, our proposed method allows us to correctly identify the salient motion sequence “adding ingredinet into a bowl”, as opposed to simply “cooking”. Similar conclusions can also be drawn from the other two examples.

On the other hand, we have also visualized the attended trajectory clusters which echoes the generated sentence description. Specifically, we maintain the trajectory cluster with highest attentional weight at each LSTM time step and



Baseline: a man is running.
Our(with Trajectory Attention): a man is kicking a ball.
Ref: a man is kicking a ball.

Baseline: a woman is cooking.
Our(with Trajectory Attention): a woman is adding ingredient into a bowl.
Ref: a woman is adding ingredient into a bowl.

Baseline: a woman is watching TV .
Our(with Trajectory Attention): a woman is lying on the bed.
Ref: a girl is lying on the bed.

Baseline: a polar bear is playing .
Our(with Trajectory Attention): a polar bear is walking on the ice.
Ref: two polar bears are walking.

Figure 2. Four sample videos and their corresponding generated and ground-truth descriptions. The middle images in each panel shows the visualization results of our attended trajectory clusters, we use the red rectangle to mark the attended clusters.

draw all the extracted trajectories on the video, with different trajectory clusters being labeled with different color. The trajectory visualization result of each sample video is also listed in Figure 2. As shown in the figure, our proposed TSA-ED framework can accurately capture the salient motion parts while generating more elaborate video captioning. For each generated statement, we can usually locate explanatory trajectory data to support our prediction. For example, when generating the caption "adding ingredient into a bowl" in the top-right of the figure, the most salient trajectory clusters attended by our algorithm is exactly the "pouring" action in the video. Indeed, the proposed attentive trajectory localization mechanism provides an effective visualization tool for video captioning and thus and can greatly enhance the model's interpretability.

5.5. Component Analysis and Comparison

Our proposed TSA-ED framework models both the local motion information and the sentence semantic structure in video captioning. To demonstrate the effectiveness of the introduction of local motion information and the significance of the structured attention mechanism, we have conducted two experiments for internal comparison. Specifically, we respectively train two models with different settings. One is an encoder-decoder network using only global frame feature and the other is our plain trajectory attention model without resorting to sentence structure information.

As shown in Table 3, the encoder-decoder network without trajectory motion information only achieve 31.2%, 64.2% and 46.6% respectively on the metric of METEOR, CIDEr and BLEU_4 metric, which deteriorate our model with trajectory attention by 5.74%, 12.9% and 7.0% respectively. Similar conclusions can be obtained in qualitative analysis and visualization results. As shown in Fig 2, the introduction of the appearance and local trajectory motion information can indeed produce more refined and accurate description generation.

As discussed in the "Decoder" section, we dynamically change the feature vectors according to the parsed sentence structure. In order to validate the effectiveness and necessity of introducing structured attention mechanism, we also compare our model with the plain attention model. As shown Table 3, incorporating sentence structure information into the traditional attention framework brings performance boost from the original model by 2.72%, 1.63% and 3.39% in terms of METEOR, CIDEr and BlUE_4, respectively.

Method	METEOR	CIDEr	BlUE_4
Encoder-Decoder without Trajectory	31.2	64.2	46.6
Encoder-Decoder with Attention Trajectory	33.1	73.7	50.1
Ours	34.0	74.9	51.7

Table 3. Component-wise efficacy of the proposed trajectory structured attentional encoder-decoder framework on MSVD dataset

6. Conclusion

In this paper, we have introduced a trajectory structured attentional encoder-decoder network which explores both the fine-grained motion information and the sentence semantic structure for video caption. Experimental results demonstrate that our proposed method can generate much more fine-grained captioning and achieve state-of-the-art performance on the public benchmarks. Moreover, the attentive trajectory localization mechanism can be regarded as an effective visualization tool and can greatly enhance the model's interpretability.

References

[1] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop*, volume 29, pages 65–72, 2005.

[2] D. Chen and C. Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750, 2014.

[3] F. Chen, R. Ji, J. Su, Y. Wu, and Y. Wu. Structcap: Structured semantic embedding for image captioning. In *ACMM*, MM '17, pages 46–54. ACM, 2017.

[4] T. Chen, Z. Wang, G. Li, and L. Lin. Recurrent attentional reinforcement learning for multi-label image recognition. *arXiv preprint arXiv:1712.07465*, 2017.

[5] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *NIPS Workshop*, number EPFL-CONF-192376, 2011.

[6] R. Fakoor, A.-r. Mohamed, M. Mitchell, S. B. Kang, and P. Kohli. Memory-augmented attention modelling for videos. *arXiv preprint arXiv:1611.02261*, 2016.

[7] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[8] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*, pages 2712–2719, 2013.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua. Hierarchical multimodal lstm for dense visual-semantic embedding. In *CVPR*, pages 1881–1889, 2017.

[12] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, pages 1029–1038, 2016.

[13] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, pages 4594–4602, 2016.

[14] Y. Pan, T. Yao, H. Li, and T. Mei. Video captioning with transferred semantic attributes. *arXiv preprint arXiv:1611.07675*, 2016.

[15] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318. Association for Computational Linguistics, 2002.

[16] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, pages 1242–1249. IEEE, 2012.

[17] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *CVPR*, pages 433–440, 2013.

[18] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, pages 510–526. Springer, 2016.

[19] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

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

[21] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence - video to text. In *ICCV*, December 2015.

[22] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[23] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176. IEEE, 2011.

[24] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013.

[25] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015.

[26] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36. Springer, 2016.

[27] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin. Multi-label image recognition by recurrently discovering attentional regions. In *ICCV*, 2017.

[28] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[29] R. Xu, C. Xiong, W. Chen, and J. J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. 2015.

[30] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *ICCV*, pages 4507–4515, 2015.

[31] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, pages 4584–4593, 2016.