Structured Two-Stream Attention Network for Video Question Answering

Lianli Gao,[1] Pengpeng Zeng,[1] Jingkuan Song,[1] Yuan-Fang Li,[2] Wu Liu,[3] Tao Mei,[3] Heng Tao Shen[1]*
[1]Center for Future Media and School of Computer Science and Engineering,
University of Electronic Science and Technology of China [2]Monash University [3]JD AI Research
lianli.gao@uestc.edu.cn, {is.pengpengzeng,jingkuan.song}@gmail.com, {liuwu,tmei}@live.com, shenhengtao@hotmail.com

Abstract

To date, visual question answering (VQA) (i.e., image QA and video QA) is still a holy grail in vision and language understanding, especially for video QA. Compared with image QA that focuses primarily on understanding the associations between image region-level details and corresponding questions, video QA requires a model to jointly reason across both spatial and long-range temporal structures of a video as well as text to provide an accurate answer. In this paper, we specifically tackle the problem of video QA by proposing a Structured Two-stream Attention network, namely STA, to answer a free-form or open-ended natural language question about the content of a given video. First, we infer rich long-range temporal structures in videos using our structured segment component and encode text features. Then, our structured two-stream attention component simultaneously localizes important visual instance, reduces the influence of background video and focuses on the relevant text. Finally, the structured two-stream fusion component incorporates different segments of query and video aware context representation and infers the answers. Experiments on the large-scale video QA dataset *TGIF-QA* show that our proposed method significantly surpasses the best counterpart (i.e., with one representation for the video input) by 13.0%, 13.5%, 11.0% and 0.3 for Action, Trans., TrameQA and Count tasks. It also outperforms the best competitor (i.e., with two representations) on the Action, Trans., TrameQA tasks by 4.1%, 4.7%, and 5.1%.

Introduction

Recently, tasks involving vision and language have attracted considerable interests. Those include captioning (Gu et al. 2018; Chen et al. 2018; Song et al. 2017; 2018a) and visual question answering (Antol et al. 2015; Gao et al. 2015; Ren, Kiros, and Zemel 2015b; Song et al. 2018b; Gao et al. 2018b). The task of captioning is to generate natural language descriptions of an image or a video. On the other hand, visual question answering (VQA) (i.e., image QA and video QA) aims to provide the correct answer to a question regard to a given image/video. It has been regarded as an important Turing test to evaluate the intelligence of a machine (Lu et al. 2018a). The VQA problem plays a significant role in various applications, including human-machine interaction and tourist assistance. However, it is a challenging task, as it is required to understand both language and vision content to consider necessary commonsense and semantic knowledge, and to finally make reasoning to obtain the correct answer.

Image QA, which aims to correctly answer a question about an image, has achieved great progress recently. Most existing methods for image QA use the attention mechanism (Antol et al. 2015), and they can be divided into two main types: visual attention and question attention. The former attention focuses on the most relevant regions to correctly answer a question by exploring their relationships, which addresses "where to look". The latter attention attends to specific words in the question about visual information, which addresses "what words to listen to". Some works jointly perform visual attention and question attention (Lu et al. 2016).

In comparison, video QA is more challenging than image QA, as videos contain both appearance and motion information. The main challenges to video QA are threefold: first, a method needs to consider long-range temporal structures without missing important information; second, the influence of video background needs to be minimized to localize the correspond video instances; third, segmented information and text information need to be well fused. Therefore, we need more sophisticated video understanding techniques that can understand frame-level visual information and the temporal coherence during the progression of the video. Video QA models also requires reasoning ability on spatial and long-range temporal structures of both video and text to infer an accurate answer.

Attention mechanisms has also been adopted for video QA, including spatial-temporal attention (Jang et al. 2017) and co-memory attention (Gao et al. 2018a). Temporal attention learns which frames in a video to attend to, which is captured as *whole-video* features. Co-memory proposes a co-memory attention mechanism: an appearance attention model to extract useful information from spatial features, and a motion attention model to extract useful cues from optical flow features. It concatenates the attended spatial and temporal features to predict the final results.

We observe that answering some questions in video QA requires focusing on many frames, which are equally important (e.g., How many times does the man step?). Using only current attention mechanisms, and hence whole-

*Heng Tao Shen and Jingkuan Song are corresponding authors.

video-level features, may ignore important frame-level information. Motivated by this observation, we introduce a new structure, namely structured segment, that divides video feature into N segments and then takes each segment as input for a shared attention model. Thus, we can obtain many important frames from multiple segments. For better linking and fusing information from both video segments and the question, we propose a Structured Two-stream Attention network (STA) to learn high-level representations. Specifically, our model has two levels of decoders, where the first-pass decoder infers rich long-range temporal structures with our structured segment, and the second-pass encoder simultaneously localizes action instance and avoids the influence of background video with the assistance of structured two-stream attention.

Our STA model achieves state-of-the-art performance on a large-scale dataset: *TGIF-QA* dataset. To summarize, our major contributions include: 1) We propose a new architecture, Structured Two-stream Attention network (STA), for the video QA task by jointly attending to both spatial and long-range temporal information of a video as well as text to provide an accurate answer. 2) The rich long-range temporal structures in videos are captured by our structured segment component, while our structured two-stream attention component can simultaneously localize action instance and avoid the influence of background video. 3) Experimental results show that our proposed method significantly outperforms the state-of-the-arts in the Action, Trans. and FrameQA tasks on the *TGIF-QA*. Notably, we represent our videos using only one type of visual features.

Related Work

Image Question Answering

Image QA (Antol et al. 2015; Gao et al. 2015; Ren, Kiros, and Zemel 2015b; Lu et al. 2018b; Nam, Ha, and Kim 2017; Yang et al. 2016; Xu and Saenko 2016; Lu et al. 2016; Patro and Namboodiri 2018; Teney et al. 2018), the task that infers answers to questions on a given image, has achieved much progress recently. Based on the framework of image captioning, most early works adopt typical CNN-RNN models, which use Convolutional Neural Networks (CNN) to extract image features and use Recurrent Neural Networks (RNN) to represent question information. They integrate image features with question features using some simple fusion methods such as concatenation, summation and element-wise multiplication. Finally, the fused features are fed into a softmax classifier to infer a correct answer. It has been observed that many questions are only related to some specific regions of an image, and various attention mechanisms has been introduced into image QA instead of using the global image features. There are two main types of attention mechanisms: visual attention and question attention. Specifically, visual attention learns which specific regions in the image to focus on for the question, while question attention attends to specific words in the question about vision information. The work in (Yang et al. 2016) designs a Stacked Attention Networks which can search question-related image regions by performing multi-step visual attention operations. In (Lu et al. 2016; Nam, Ha, and Kim 2017), they present a co-attention mechanism that jointly performs question-guided visual attention and image-guided question attention to address the 'which regions to look' and 'what words to listen to' problems respectively. The typically used, simple fusion methods (e.g., concatenation, summation and element-wise multiplication) on visual and textual features cannot sufficiently exploit the relationship between images and questions. To tackle this problem, some researchers introduced more sophisticated fusion strategies. Bilinear (pooling) method (Gao et al. 2016) is one of the pioneering works to efficiently and expressively combine multimodal features by using an outer product of two vectors. Based on MCB (Gao et al. 2016), lots of variants have been proposed, including MLB (Kim et al. 2016) and MFB (Yu et al. 2017b). The work in (Nguyen and Okatani 2018) proposes a dense co-attention network (DCN) that computes an affinity matrix to obtain more fine-grained interactions between an image and a question.

Video Question Answering

Compared with image QA, video QA is more challenging. The LSMDC-QA dataset (Rohrbach et al. 2017) introduced the movie fill-in-the-blank task by transforming the LSMDC movie description dataset to the video QA domain. The MovieQA dataset (Tapaswi et al. 2016) aims to evaluate automatic story comprehension from both videos and movie scripts. The work in (Jang et al. 2017) introduces a new large-scale dataset named *TGIF-QA* and designs three new tasks specifically for video QA. The attention mechanism has also been widely used in video QA. The work in (Yu et al. 2017a) proposes a semantic attention mechanism, which detects concepts from the video first and then fuses them with text encoding/decoding to infer an answer. The work in (Jang et al. 2017) proposes a dual-LSTM based approach with both spatial and temporal attention. The spatial attention mechanism uses the text to focus attention over specific regions in an image. The temporal attention mechanism guides which frames in the video to look at for answering the question. The work in (Gao et al. 2018a) proposes a co-memory attention network for video QA, which jointly models motion and appearance information to generate attention on both domains. They introduce a method called dynamic fact ensemble to dynamically produce temporal facts in each cycle of fact encoding. These methods usually extract compact whole-video-level features, while not adequately preserving frame-level information. However, a question to a video might be relevant to a sequence of frames, e.g., 'How many times does the man step'. The compact whole-video-level features are not as informative as more fine-grained frame-level features. To address this issue, we propose to split a video into multiple segments in order to achieve a better balance between information compactness and completeness. We introduce our method in the next section.

Methodology

Recall that our aim is to efficiently extract video spatial and long-range temporal structures and then improve fu-

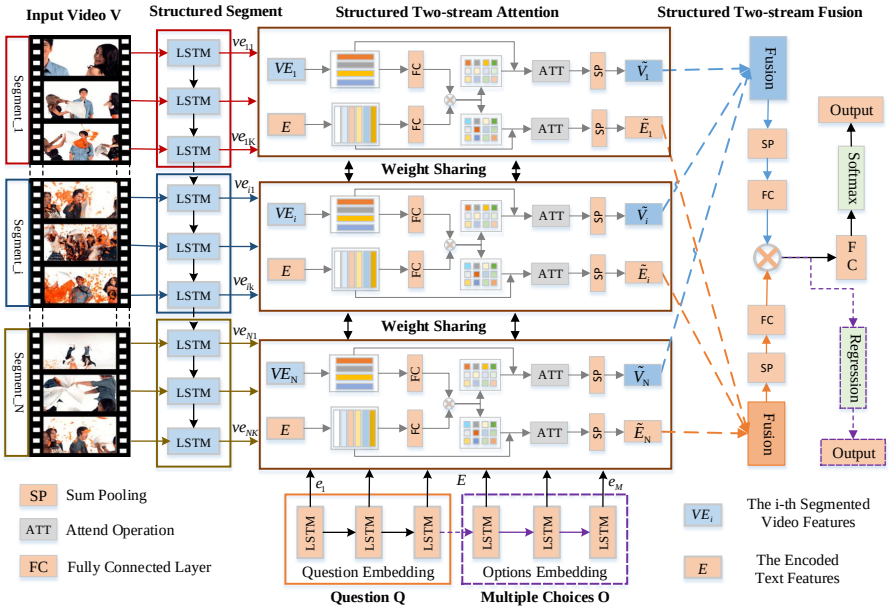

Figure 1: The framework of our proposed Structured Two-stream Attention Network (STA) for video QA.

sion of video and text representations to provide an accurate answer. As discussed in the Introduction, the primary challenges are threefold: (1) the incorporation of long-range temporal structures without missing important information; (2) the minimization of the influence of video background to localize the correspond video instances; and (3) the adequate fusion of segmented information with text information.

Our proposed framework is shown in Fig.1. Formally, the input is a video V, a question Q and a set of answer options O. In addition, only multiple choice type questions require the input of O, as shown in the purple dashed box in Fig.1. Specifically, our framework consists of a number of *a structured segments* that focuses on obtaining video long-range temporal information, *a structured two-stream attention* that fuses language and video visual features repeatedly, on top of which is *a structured two-stream fusion based answer prediction module* that fuses multi-modal segmental representations to predict answers. Below, we present the details of the above three major components.

Structured Segment

Video Feature Extraction. Following previous work (Gao et al. 2018a; Yu et al. 2017a), we employ Resnet-152 (He et al. 2016), pre-trained on the ImageNet 2012 classification dataset (Russakovsky et al. 2014), to extract video frame appearance features. More feature pre-processing details are given in Experiments section. For each video frame, we obtain a 2048-D feature vector, which is extracted from the pool5 layer and represents the global information of that frame. Therefore, an input video can be represented as:

$$V = [v_1, v_2, ..., v_T], v_t \in \mathbb{R}^{2048} \quad (1)$$

where T is the length of a video.

For encoding sequential data streams, Recurrent Neural Networks (RNN) are widely and successfully used, especially in machine translation research. In this paper, we employ Long Short-term Memory (LSTM) networks to further encode video features $\{\mathrm{v_t}\}_{\mathrm{t=1}}^{\mathrm{T}}$ to extract useful cues. For each step, e.g., t-th step, the LSTM unit takes the t-th frame features and the previous hidden state $\mathrm{h}_{\mathrm{t-1}}^{\mathrm{v}}$ as inputs to output the t-th hidden state $\mathrm{h}_{\mathrm{t}}^{\mathrm{v}} \in \mathbb{R}^D$, where we set the dimension D = 512.

$$\mathrm{h}_{\mathrm{t}}^{\mathrm{v}} = \mathrm{LSTM}(\mathrm{v}_t, \mathrm{h}_{t-1}^{v}) \quad (2)$$

Structured Segment. Previous work such as TGIF-QA (Jang et al. 2017) adopts a dual-layer LSTM to encode video features and then concatenates the last two hidden states of the dual-layer LSTM to represent whole-video-level information. This poses a risk of missing important frame-level information. To solve this problem, we introduce a new structure, namely structured segment, that firstly utilizes one-layer LSTMs to obtain T hidden states ($\{\mathrm{h}_{\mathrm{t}}^{v}\}_{\mathrm{t=1}}^{\mathrm{T}}$) and then divides the T hidden states into N segments ($\{VE_i\}_{i=1}^{N}$). After this stage, a video can be represented as $\{VE_i\}_{i=1}^{N}$, and the i-th segment VE_i is represented as:

$$VE_i = \{ve_{i1}, ve_{i2}, ..., ve_{iK}\}, ve_{ik} \in \mathbb{R}^{512} \quad (3)$$

where ve_{ik} is the hidden states of the k-th frame in the i-th segment, K the total number of hidden states for each segment. The value of K is the same for each segment.

Text Encoder

For our video QA task, there are two types of questions: open-ended question and multiple-choice question. For the first type, our framework takes only the question as the text input, while for the second type, our framework takes both

the question and answer options as the text input. The final text feature is represented as E.

Question Encoding. A question, consisting of M words, is first converted into a sequence $Q = \{q_m\}_{m=1}^{M}$, where q_m is a one-hot vector representing the word at position m. Next, we employ the word embedding GloVe (Pennington, Socher, and Manning 2014) pre-trained on the Common Crawl dataset, to process each word to obtain a fixed word vector. After GloVe, the m-th word is represented as x_m^q. Next, we utilize a one-layer LSTM on top of the word embeddings to model the temporal interactions between words. The LSTM takes the embedding vectors $\{x_m^q\}_{m=1}^{M}$ as inputs, and finally we obtain a question feature E for answer prediction process. As a result, the question encoding process can be defined as below:

$$x_m^q = W_e^q q_m \quad (4)$$

$$e_m^q = LSTM(x_{\mathrm{m}}^{\mathrm{q}}, e_{m-1}^q) \quad (5)$$

where W_e^q is an embedding matrix. The dimension of all the LSTM hidden states is set to $D = 512$. Finally, after question encoder, Q is represented as $E = \{e_1^q, \cdots, e_M^q\}$.

Multi-choice Encoding. For the task of Multi-choice, the input involves a question and a set of answer candidates. To process answer candidates, we follow the above question encoding procedure to transform each word of the options into a one-hot vector and then further embed it the GloVe. We consider answer candidates as complementary to the question. Therefore, the text input of our framework becomes $Q' = [Q, O]$, where O is the answer candidate features and $[,]$ represents concatenation. Furthermore, the one-layer LSTM unit takes the merged Q' as the input to extract text feature E. We formulate this encoding process as below:

$$Q' = [Q, O] = \{q_1', \cdots, q_M'\} \quad (6)$$

$$x_m^q = W_e^q q_m' \quad (7)$$

$$e_m^q = LSTM\left(x_m^q, e_{m-1}^q\right) \quad (8)$$

where M is the sum of the length of question words and the length of all candidate words. Finally, after the multi-choice encoder, Q' is represented as $E = \{e_1^q, \cdots, e_M^q\}$.

Structured Two-stream Attention Module

We now describe the second major component, the Structured Two-stream Attention layer (seen Fig.1), which links and fuses information from both video segments and text. This attention layer consists of N two-stream (i.e., text and video features) attentions and all the attention models share parameters.

For the i-th two-stream attention model, it takes the i-th segmented video encoded feature VE_i and text feature E as input to learn interactions between them to update both VE_i and E. Here, we denote the input to the i-th two-stream attention by $VE_i = \{ve_{i1}, \cdots, ve_{iK}\} \in \mathbb{R}^{D \times K}$ and $E = \{e_1, \cdots, e_M\} \in \mathbb{R}^{D \times M}$. Unlike previous video QA methods such as TIGF-QA (Gao et al. 2018a) and Co-memory (Gao et al. 2018a), which simply concatenate video frame features with text question features to form a new feature for answer prediction, our two-stream attention mechanism calculates attention in two directions: from video to question as well as from question to video. Both attention scores are computed from a shared affinity matrix A_i, which is computed by:

$$A_i = (VE_i)^T W_s E \quad (9)$$

where W_s is a learnable weight matrix. For the convenience of calculation, we replace Eq.(9) by two separate linear projections. Thus, Eq.(9) is re-defined as below:

$$A_i = \left(W_v VE_i\right)^T \left(W_q E\right) \quad (10)$$

where W_v and W_q are linear function parameters. In essence, A_i encodes the similarity between its two inputs VE_i and E. With A_i, we can compute attentions and then attend to the two-stream features in both directions.

1st-stream: Visual Attention. Visual attention vector indicates *which frames in a video shot to attend to or most relevant to each question word*. Given $A_i \in \mathbb{R}^{K \times M}$, the attention vector is computed by the following function:

$$C_i = soft\max(\max_{col}(A_i)^T) \quad (11)$$

where $C_i \in \mathbb{R}^{K \times 1}$, $\max_{col}$ indicates column-wise max operation on A_i. After column-wise max operation, we use *softmax* operation to normalize the value to produce the attention vector C_i. Specifically, $\sum_{k=1}^{K} c_{ik} = 1$. Next, we conduct the following operation to obtain the attend video feature $\tilde{V}_i$:

$$\tilde{V}_i = \sum\nolimits_{k=1}^{K} c_{ik} ve_{ik} \quad (12)$$

where $\tilde{V}_i \in \mathbb{R}^{1 \times D}$, which contains the attended visual vectors respect to the entire question.

2nd-stream: Text Attention. Textual attention vector indicates *which question word to attend to*. Given $A_i \in \mathbb{R}^{K \times M}$, we normalize A_i in row-wise with *softmax* to derive the attention map on question words conditioned by each video frame. Formally, the attention vector is computed by the following function:

$$B_i = softmax\left(A_i\right) \quad (13)$$

where $B_i \in \mathbb{R}^{K \times M}$. With the generated attention map B_i, we utilize it to attend to the question words to produce a more representative question feature $\tilde{E}_i$. Formally, $\tilde{E}_i$ is computed by the following function:

$$\tilde{E}_i = B_i E^T \quad (14)$$

where $\tilde{E}_i \in \mathbb{R}^{K \times D}$. Finally, we conduct column-wise sum to obtain the final $\tilde{E}_i \in \mathbb{R}^{1 \times D}$. Hence the generated $\tilde{E}$ contains the attended question words vectors for the entire video segments.

Structured Two-stream Fusion

After computing the attended feature representations $\tilde{V}$ and $\tilde{E}$, where $\tilde{V} = \left\{\tilde{V}_1, \cdots, \tilde{V}_N\right\}$ and $\tilde{E} = \left\{\tilde{E}_1, \cdots, \tilde{E}_N\right\}$, we first fuse them respectively.

$$\tilde{V} = \sum\nolimits_{i=1}^{N} \tilde{\mathrm{V}}_i \quad (15)$$

$$\tilde{E} = \sum\nolimits_{i=1}^{N} \tilde{E}_i \quad (16)$$

Followed by a sum pooling, the attended video vector $\tilde{V}$ and attended question vector $\tilde{E}$ are pooled and combined together to yield H for answer prediction.

$$H = Relu(W_{fv}\tilde{V} + b_v) \otimes Relu(W_{fq}\tilde{E} + b_q) \quad (17)$$

where W_{fv} and W_{fq} are parameters; b_v and b_q are bias terms; $\otimes$ is the element-wise multiplication, $Relu$ is the activation function.

Answer Decoder Moduler

Our output layer (i.e. answer decoder) is application-specific. Our framework allows us to easily swap the output layer based on the task type with the rest of the architecture remaining exactly the same. Following previous work (Jang et al. 2017; Gao et al. 2018a), we treat the four tasks (i.e. Count, Action, State Trans., and FrameQA) in the *TGIF-QA* dataset as three different type decoders: multiple choice, open-ended numbers and open-ended words. Here, we describe the three types of answer decoders for each specific video QA task.

Multiple Choice. For the TGIF-QA dataset, both State Trans. and Action tasks belong to the multiple choice category QA. In order to solve both tasks, we apply a linear regression function for the above final output H and derive a real-valued score for each answer option.

$$s = W_h{}'H \quad (18)$$

where W_h' are weight parameters. We define s_p and s_n as the real-valued scores derived from the positive and negative answers, respectively. In order to train our mode, we minimize the hinge loss of pair comparisons, $\max(0, 1 + s_n - s_p)$.

Open-ended Numbers. For the TGIF-QA dataset, the Count task requires a model to count the number of repetitions of an action, and the answer ranges from 0 to 10. For this task, we define a linear regression function which takes a predicted output H as input and produces an integer-valued answer, ranging from 0 to 10. The output score is defined as:

$$s = [W'_h H + b_h] \quad (19)$$

where $[.]$ means rounding, W'_h are model parameters and b_h is the bias. In order to train the model, we adopt the Mean Square Error (MSE) loss between the real value and the predicted value.

Open-ended Words. Similar to the task of image QA, we treat video FrameQA as a multi-class classification problem, in which each class corresponds to a distinct answer (i.e., a dictionary word, a type of object, color, number or location). For video QA, we use a softmax classifier to infer the final answer. The candidate with the highest probability is considered as the final answer. This output is defined as:

$$o = softmax(W'_h H + b_h) \quad (20)$$

where W'_h are weight parameters and b_h is the bias. The model is optimized by minimizing the cross-entropy loss.

Note that, in this paper we deal with four tasks. For each task, we separately train each model with the corresponding answer decoder and loss function mentioned above. Eventually, each model is evaluated separately.

Table 1: Statistics of the TGIF-QA dataset.

Task	Train	Test	Total
Repetition Count	26,843	3,554	30,397
Repeating Action	20,475	2,274	22,749
State Transition	52,704	6,232	58,936
Frame QA	39,392	13,691	53,083
Total	139,414	25,751	165,165

Evaluation

In this section, we first describe the dataset, evaluation metrics and implementation details. Then we report and analyze the experimental results.

Dataset and Evaluation Settings

To evaluate the performance of the video QA models, we follow two recent video QA work (Jang et al. 2017; Gao et al. 2018a) to evaluate our method on the large-scale public video QA dataset TGIF-QA.

TGIF-QA Dataset. It is a large-scale dataset collected by (Jang et al. 2017), which is designed specifically for video QA to better evaluate a model's capacity for deeper video understanding and reasoning. Jang *et al.* collected 165k QA pairs from 71k animated Tumblr GIFs (Li et al. 2016) and defined four task types: repetition count (Count), repeating action (Action), state transition (Trans.) and video frame QA (FrameQA). Each of them has approximately 30k, 22k, 58k and 53k questions, respectively. The specific setting (e.g., train and test splits) are demonstrated in Tab.1, following the original setting of (Jang et al. 2017). Compared with FrameQA which can be answered by analyzing the content of one particular video frame, other three tasks Count, Action and Trans. can only be answered accurately only via reasoning across multiple frames. In addition, both Count and FrameQA contain open-ended questions, while Action and Trans. contain multi-choice questions.

Four Tasks Settings. In this paper, we deal with four specific video QA tasks: Count, Action, Trans. and FrameQA. Specifically, each Count question has 11 possible answers, ranging from zero to ten. It requires a model to calculate how many times an action has been repeated. For instance, "How many times does the woman chew food?". Compared with Count, Action focuses on the actions appearing in a video. Instead of asking how many times an action is performed, it asks which action is repeated a given number of times in a video. For example, "What does the woman do 4 times?". For Trans., each question provides five candidate options and all the questions are about the understanding of the transition of two action states in a video. For instance, "What does the woman do before places hands in lap?". For FrameQA, it is similar to image QA, but more complex. To provide an accurate answer, a model must locate the specific frame and its specific regions that the question refers to. For example, "what is the color of the hair?". We treat FrameQA as a multi-class classification problem.

Evaluation Metric. Following (Jang et al. 2017; Gao et al. 2018a), we use the same evaluation metrics. Specifically,

Table 2: Ablation study on the *TGIF-QA* dataset. For evaluation metric, Action, Trans. and FrameQA use ACC (%), while Count adopts Mean Square Error (MSE). V and T indicates visual attention and text attention. N indicates the number of video segments. ST is the best published methods using the same features as our model uses.

Method	Action	Trans	Frame	Count
STA-V-T(N=1)	71.94	78.67	56.01	4.26
STA-V-T(N=2)	72.16	78.84	55.92	4.25
STA-V-T(N=3)	**72.38**	78.96	56.52	4.27
STA-V-T(N=4)	72.34	**79.03**	**56.64**	**4.25**
STA-V(N=1)	71.02	77.24	54.89	4.32
STA-V(N=2)	71.46	77.47	55.05	4.37
STA-V(N=3)	71.68	77.31	55.50	4.34
STA-V(N=4)	71.24	77.57	55.47	4.33
ST (Jang et al. 2017)	59.04	65.56	45.60	4.55

FrameQA, Action and Trans., the accuracy (ACC.) is employed to evaluate model performance. Thus, the higher the ACC. is, the better the model is. The performance of Count is measured by Mean Square Error (MSE) between the true answer and the predicted answer. Therefore, lower MSE value indicates better performance of the model.

Implementation Details

For fair comparisons with recent work (Jang et al. 2017; Gao et al. 2018a), we use the same network ResNet152 (He et al. 2016) pre-trained on ImageNet 2012 classification to extract frame features. More specifically, all the frame features are obtained from the same pooling layer (pool5) and their dimension is 2048. In all tables in the experimental section, we use *R* to indicates that the input video's feature is extracted from ResNet152 feature. In addition, to reduce redundant information and reduce computation cost, we sample 36 frames from each video with equal spacing.

For text representation, we first encode each word with a pre-trained GloVe embedding to generate a 300-D vector following (Jang et al. 2017; Gao et al. 2018a). All the words are further encoded by a one-layer LSTM, whose hidden state has the size of 512. All the hidden states are concatenated and used for co-attention.

Training Details. In our experiments, the optimization algorithm is Adamax. The batch size is set as 128. The train epoch is set as 30. In addition, gradient clipping, weight normalization and dropout are employed in training. In addition, our implementation is based on the Pytorch library.

Ablation Study

The framework of our proposed STA consists of multiple major components. In order to evaluate the contribution of each component to the final performance, we conduct several ablation studies on the *TGIF-QA* dataset. Experimental results are shown in Tab. 2.

The Role of Structured Segmentation. The first block of Tab. 2 shows the effect of N, which is the number of structured segments. From the first block, we found that $N = 4$ improves performance of all four tasks to a certain extent. One possible reason is that dividing a video into multiple segments to conduct attention has the potential to locate the most relevant frames as well as to learn the long structures. However, the performance improvement with the increase in N is minor and the reason might be the nature of the GIF videos, which are well segmented and carefully curated, with an average length of 47 frames. Thus the advantages of our structured segment are not fully exploited.

Table 3: Comparison with the state-of-the-art method on the *TGIF-QA* dataset. R indicates ResNet152 features. For video representation, all methods take video spatial vectors only as the visual inputs.

Model	Action	Trans.	Frame	Count
VIS+LSTM(agg)(R)	46.8	56.9	34.6	5.09
VIS+LSTM(avg)(R)	48.8	34.8	35.0	4.80
VQA-MCB(agg)(R)	58.9	34.3	25.7	5.17
VQA-MCB(avg)(R)	29.1	33.0	15.5	5.54
CT-SAN(R)	56.1	64.0	39.6	5.13
ST(R)	59.0	65.5	45.6	4.55
STA(R)	**72.3**	**79.0**	**56.6**	**4.25**

Table 4: Comparison with the state-of-the-art multi-feature based methods on the *TGIF-QA* dataset. R, C and F indicate Resnet152, C3D and Optical Flow features, respectively. Sp and Tp indicate spatial attention and temporal attention respectively.

Model	Action	Trans	Frame	Count
ST (R+C)	60.1	65.7	48.2	4.38
ST-Sp (R+C)	57.3	63.7	45.5	4.28
ST-Sp-Tp (R+C)	57.0	59.6	47.8	4.56
ST-Tp (R+C)	60.8	67.1	49.3	4.40
ST-Tp (R+F)	62.9	69.4	49.5	4.32
Co-memory (R+F)	68.2	74.3	51.5	**4.10**
STA (R)	**72.3**	**79.0**	**56.6**	4.25

The Role of Two-stream Attention. To analyze the contribution of two-stream attention: 1st-stream visual attention and 2nd-stream text attention, we conduct the second ablation study by removing the text attention and keeping the visual attention, which is represented as STA-V in Tab.2. From the table, we can see that with the same setting of N, STA-V-T performs better than STA-V. When $N = 4$, STA-V-T surpasses STA-V by 1.1% on Action, 1.46% on Trans. and 1.17% on FrameQA, and reduces MSE to 4.25 on Count. This ablation study shows the beneficial effects of our text attention.

In order to examine the influence of the visual attention, we compare our STA-V with the best published results obtained by the spatial-temporal reasoning (ST) method (Jang et al. 2017). From the last block we can see that our STA-V significantly outperforms ST by a large margin (12.24%, 12.07%, 9.87% on Action, Trans. and FrameQA, respectively). In addition, for Count, compared with ST with MSE of 4.55, STA-V-T reduces the error score to 4.33.

Qualitative Results

To understand the effect of our attention mechanism, we show some examples in Fig. 2. The first row demonstrates

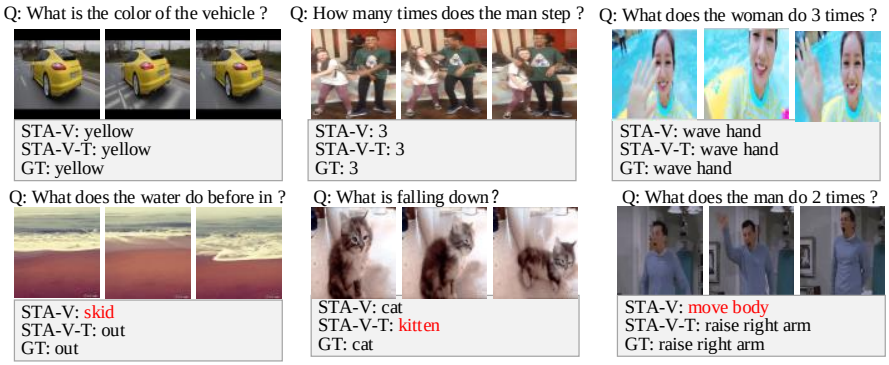

Figure 2: Some qualitative results produced by our models. Text in orange indicates incorrect answers.

three positive examples, where both models (i.e., STA-V and STA-V-T) can provide correct answers. The second row shows other examples. More specifically, the first and the third examples show that STA-V-T answers the questions correctly, while STA-V fails. The middle examples shows that occasionally STA-V can provide the correct answer.

Comparing with State-of-the-Arts on the TGIF-QA Dataset

To date, a few studies have been conducted on the video QA task. Existing methods can be divided into two categories: 1) image-based approaches including VIT+LSTM (Ren, Kiros, and Zemel 2015a) and VQA-MCB (Fukui et al. 2016); and 2) video-based approaches including ST (Jang et al. 2017) and Co-memory (Gao et al. 2018a). Our method belongs to the second category. Usually, videos contain two types of features: spatial and temporal. Spatial features are usually extracted from a CNN network, while temporal features are extracted from optical flows or via a C3D network. Therefore, in this case, we can divide the existing methods into two categories: single feature based methods and multi-feature based methods.

Our method STA utilizes single feature for representing videos. Therefore, in this section, we first compare our method with existing methods which take only ResNet152 frame features as visual inputs to conduct answer prediction. The experimental results are shown in Tab. 3. In order to apply image-based approach, multiple frames' spatial features must be firstly merged into a vector, which is considered as the alternative of image features to be fed into an image-based agg. or avg. based model. Agg. and avg. are proposed in (Jang et al. 2017) to directly apply image-based methods. The agg. is conducted by averaging all frames' features and uses the average spatial feature as input to the model. Compared with agg., avg. is more complicated. It runs answer prediction on each frame, and then averages all frames' predicted answers to obtain the final result. It can be seen from Tab. 3 that our method outperforms the best publish results ST(R) by 13.3%, 13.5% and 11.0% on Action, Trans. and FrameQA, respectively. For count, STA reduces the error score to 4.25.

Furthermore, we compare our method STA with multi-feature based methods, which include ST (Jang et al. 2017) and Co-memory (Gao et al. 2018a). Specifically, ST combines ResNet152 features (marked as R in Tab.4) with C3D features (C) or optical flow features (F), while Co-memory incorporates both R and F. The comparison results are given in Tab. 4. From these results, we can observe that Co-memory (R+F) outperforms all variants of ST, and achieves the best performance on the Count task. Compared with Co-memory (R+F), our STA model only takes ResNet152 features as input. However, for Action, Trans. and FrameQA tasks, it significantly outperforms Co-memory (R+F) (68.2%, 74.3%, and 51.5%, respectively) and yields the best performance reaching 72.3%, 79.0%, 56.6%, respectively. Even for the Count task, our STA achieves the second best and comparable result.

Conclusion

In this work, we propose a novel Structured Two-stream Attention Network (STA) for Video Question Answering. We first utilize our structured segment component to infer rich long-range temporal structures in videos and also encode questions to features. Then, instead of simply concatenating video and question features to answer the questions, we utilize a two-stream attention mechanism to simultaneously localize visual instances relevant to the questions and to avoid the influence of background video or irrelevant text. Finally, the structured two-stream fusion component incorporates different segments of query- and video-aware context representations and infers the answers for different types of questions. Our comprehensive evaluation on the *TGIF-QA* dataset shows our STA model outperforms state-of-the-art methods significantly.

Acknowledgements

This work is supported by the Fundamental Research Funds for the Central Universities (Grant No. ZYGX2014J063, No. ZYGX2016J085), the National Natural Science Foundation of China (Grant No. 61772116, No. 61502080, No. 61632007, No. 61602049), and JD AI Research.

References

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: visual question answering. In *ICCV*, 2425–2433.

Chen, H.; Ding, G.; Zhao, S.; and Han, J. 2018. Temporal-difference learning with sampling baseline for image captioning. In *AAAI*.

Fukui, A.; Park, D. H.; Yang, D.; Rohrbach, A.; Darrell, T.; and Rohrbach, M. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 457–468.

Gao, H.; Mao, J.; Zhou, J.; Huang, Z.; Wang, L.; and Xu, W. 2015. Are you talking to a machine? dataset and methods for multilingual image question answering. *CoRR* abs/1505.05612.

Gao, Y.; Beijbom, O.; Zhang, N.; and Darrell, T. 2016. Compact bilinear pooling. In *CVPR*, 317–326.

Gao, J.; Ge, R.; Chen, K.; and Nevatia, R. 2018a. Motion-appearance co-memory networks for video question answering. *CoRR* abs/1803.10906.

Gao, L.; Zeng, P.; Song, J.; Liu, X.; and Shen, H. T. 2018b. Examine before you answer: Multi-task learning with adaptive-attentions for multiple-choice VQA. In *ACM MM*, 1742–1750.

Gu, J.; Cai, J.; Wang, G.; and Chen, T. 2018. Stack-captioning: Coarse-to-fine learning for image captioning. In *AAAI*.

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

Jang, Y.; Song, Y.; Yu, Y.; Kim, Y.; and Kim, G. 2017. TGIF-QA: toward spatio-temporal reasoning in visual question answering. In *CVPR*, 1359–1367.

Kim, J.; On, K. W.; Lim, W.; Kim, J.; Ha, J.; and Zhang, B. 2016. Hadamard product for low-rank bilinear pooling. *CoRR* abs/1610.04325.

Li, Y.; Song, Y.; Cao, L.; Tetreault, J. R.; Goldberg, L.; Jaimes, A.; and Luo, J. 2016. TGIF: A new dataset and benchmark on animated GIF description. In *CVPR*, 4641–4650.

Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2016. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 289–297.

Lu, P.; Ji, L.; Zhang, W.; Duan, N.; Zhou, M.; and Wang, J. 2018a. R-VQA: learning visual relation facts with semantic attention for visual question answering. In *SIGKDD*, 1880–1889.

Lu, P.; Li, H.; Zhang, W.; Wang, J.; and Wang, X. 2018b. Co-attending free-form regions and detections with multi-modal multiplicative feature embedding for visual question answering. In *AAAI*.

Nam, H.; Ha, J.; and Kim, J. 2017. Dual attention networks for multimodal reasoning and matching. In *CVPR*, 2156–2164.

Nguyen, D.-K., and Okatani, T. 2018. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *CVPR*.

Patro, B., and Namboodiri, V. P. 2018. Differential attention for visual question answering. In *CVPR*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.

Ren, M.; Kiros, R.; and Zemel, R. S. 2015a. Exploring models and data for image question answering. In *NIPS*, 2953–2961.

Ren, M.; Kiros, R.; and Zemel, R. S. 2015b. Image question answering: A visual semantic embedding model and a new dataset. *CoRR* abs/1505.02074.

Rohrbach, A.; Torabi, A.; Rohrbach, M.; Tandon, N.; Pal, C. J.; Larochelle, H.; Courville, A. C.; and Schiele, B. 2017. Movie description. *International Journal of Computer Vision* 123(1).

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. S.; Berg, A. C.; and Li, F. 2014. Imagenet large scale visual recognition challenge. *CoRR* abs/1409.0575.

Song, J.; Gao, L.; Guo, Z.; Liu, W.; Zhang, D.; and Shen, H. T. 2017. Hierarchical LSTM with adjusted temporal attention for video captioning. In *IJCAI*, 2737–2743.

Song, J.; Guo, Y.; Gao, L.; Li, X.; Hanjalic, A.; and Shen, H. T. 2018a. From deterministic to generative: Multimodal stochastic rnns for video captioning. *IEEE Transactions on Neural Networks and Learning Systems* 1–12.

Song, J.; Zeng, P.; Gao, L.; and Shen, H. T. 2018b. From pixels to objects: Cubic visual attention for visual question answering. In *IJCAI*, 906–912.

Tapaswi, M.; Zhu, Y.; Stiefelhagen, R.; Torralba, A.; Urtasun, R.; and Fidler, S. 2016. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 4631–4640.

Teney, D.; Anderson, P.; He, X.; and van den Hengel, A. 2018. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *CVPR*.

Xu, H., and Saenko, K. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 451–466.

Yang, Z.; He, X.; Gao, J.; Deng, L.; and Smola, A. J. 2016. Stacked attention networks for image question answering. In *CVPR*, 21–29.

Yu, Y.; Ko, H.; Choi, J.; and Kim, G. 2017a. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, 3261–3269.

Yu, Z.; Yu, J.; Fan, J.; and Tao, D. 2017b. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *ICCV*, 1839–1848.