Open-Ended Long-Form Video Question Answering via Hierarchical Convolutional Self-Attention Networks

Zhu Zhang\textsuperscript{1}, Zhou Zhao\textsuperscript{*1}, Zhijie Lin\textsuperscript{1}, Jingkuan Song\textsuperscript{2} and Xiaofei He\textsuperscript{3}

\textsuperscript{1}College of Computer Science, Zhejiang University, China
\textsuperscript{2}University of Electronic Science and Technology of China, China
\textsuperscript{3}State Key Lab of CAD&CG, Zhejiang University, China
\{zhangzhu, zhaozhou, linzhijie\}@zju.edu.cn, \{jingkuan.song, xiaofeihe\}@gmail.com

Abstract

Open-ended video question answering aims to automatically generate the natural-language answer from referenced video contents according to the given question. Currently, most existing approaches focus on short-form video question answering with multi-modal recurrent encoder-decoder networks. Although these works have achieved promising performance, they may still be ineffectively applied to long-form video question answering due to the lack of long-range dependency modeling and the suffering from the heavy computational cost. To tackle these problems, we propose a fast Hierarchical Convolutional Self-Attention encoder-decoder network (HCSA). Concretely, we first develop a hierarchical convolutional self-attention encoder to efficiently model long-form video contents, which builds the hierarchical structure for video sequences and captures question-aware long-range dependencies from video context. We then devise a multi-scale attentive decoder to incorporate multi-layer video representations for answer generation, which avoids the information missing of the top encoder layer. The extensive experiments show the effectiveness and efficiency of our method.

1 Introduction

Open-ended video question answering is an important problem in visual information retrieval, which automatically generates the accurate answer from referenced video contents according to the given question. Most existing works employ the multi-modal recurrent encoder-decoder framework, which first encodes the multi-modal video and question contents into a joint representation, and then generates the natural-language answer [Xue et al., 2017; Zhao et al., 2018]. Although these approaches have achieved excellent performance in short-form video question answering, they may still be ineffectively applied to the long-form video question answering due to the lack of long-range dependency modeling and the suffering from the heavy computational cost.

\textsuperscript{*}Zhou Zhao is the corresponding author.
Hierarchical Convolutional Self-Attention Encoder-Decoder Networks

3.1 The Problem

We present a video as a sequence of frames \( v = \{ v_i \}_{i=1}^{n} \in V \), where \( v_i \) is the feature of the \( i \)-th frame and \( n \) is the feature number. Each video is associated with a natural language question, denoted by \( q = \{ q_i \}_{i=1}^{m} \in Q \), where \( q_i \) is the feature of the \( i \)-th word and \( m \) is the word number. And the ground-truth answer for each question is denoted by \( a = \{ a_i \}_{i=1}^{r} \in A \) of length \( r \), where \( a_i \) is the \( i \)-th word token. Our goal is to generate a natural-language answer from referenced long-form video contents according to the given question. The overall framework is shown in Figure 2.

3.2 Hierarchical Convolutional Self-Attention Encoder

In this section, we introduce the hierarchical convolutional self-attention encoder that unifies the convolutional sequence modeling, attentive segmentation and question-aware self-attention mechanism into a common framework. We first extract the visual features \( v = \{ v_i \}_{i=1}^{n} \) using a pre-trained 3D-ConvNet [Tran et al., 2015] and then apply a linear projection for dimensionality reduction. Compared with RNN-based encoder, convolutional encoder lacks precise position modeling. Hence, we add position encoding [Vaswani et al., 2017] into the initial video sequence to supplement the temporal information. After that, we employ a pre-trained word2vec [Mikolov et al., 2013] to extract the word features \( q = \{ q_i \}_{i=1}^{m} \) and then develop a bi-directional GRU networks (BiGRU) to learn the question semantic representations. The BiGRU networks incorporate contextual in-
What...girldo...?
C3D
BiGRU
Multi-ScaleAttention
ConvUnit
GLU
Add
ConvUnit
AttentiveSegmentationUnit
... for the $i$-th segment ($ol_2^{i,1}; \cdots; ol_2^{i,H}$) with $H$ elements, we compute the attention weight of $j$-th

for the $i$-th segment ($ol_2^{i,1}; \cdots; ol_2^{i,H}$) with $H$ elements, we compute the attention weight of $j$-th

formation for each word, given by

$$h_i^f = GRU_f^f(q_i, h_{i-1}^f),$$
$$h_i^b = GRU_b^b(q_i, h_{i+1}^b),$$
$$h_i^q = [h_i^f; h_i^b],$$
$$h_i^Q = [h_i^m; h_i^b],$$

where $GRU_f^f$ and $GRU_b^b$ represent the forward and backward GRU, respectively. And the contextual representation $h_i^q$ is the concatenation of the forward and backward hidden state at the $i$-th step. Thus, we get the question semantic representations $h_q^{l-1} = (h_1^q, h_2^q, \ldots, h_n^q)$ and global question representation $h_Q^Q$.

Next, we introduce the hierarchical convolutional self-attention encoder, composed by $L$ convolutional self-attention layers. The $l$-th layer takes the sequence representations $h_i^{l-1} = (h_i^{l-1,1}; h_i^{l-1,2}; \cdots; h_i^{l-1,n_l})$ of length $n_l$ as input, and output another sequence representations $h_i^{l} = (h_1^l, h_2^l, \ldots, h_n^l)$ of length $n_l$, where $n_{l-1} = H \times n_l$ and the sequence is reduced by a factor of $1/H$ per layer. The input of the first layer is the initial video sequence $v = \{v_i\}_{i=1}^n$ with $n$ elements. Concretely, each layer consists of two convolution units, an attentive segmentation unit and a question-aware self-attention unit.

**Convolution Unit**

The convolution unit can efficiently model the local context of long-form video contents. We first conduct a one-dimensional convolution for the input sequence. Specifically, we parameterize the convolution kernel by $W^l \in \mathbb{R}^{kd \times d}$ and $b^l \in \mathbb{R}^{kd}$, where $k$ is the kernel width and $d$ represents the dimensionality of sequence elements. We then concatenate successive $k$ elements and map them into a single output $Y \in \mathbb{R}^{kd}$ with the convolutional kernel, given by

$$Y = W^l[h_{i-k/2}^{l-1}; \cdots; h_{i+k/2}^{l-1}] + b^l.$$ We pad the input sequences by $k/2$ zero vectors on the both sides to keep the sequence length invariable.

We then devise a special non-linearity operation by the gated linear unit, called as GLU [Gehring et al., 2017], on the output $Y = [A; B] \in \mathbb{R}^{kd}$:

$$\text{GLU}(A; B) = A \odot \delta(B),$$

where $A, B \in \mathbb{R}^{d}$ are the inputs to the GLU, $\delta$ is the sigmoid function, $\otimes$ represents the point-wise multiplication and the output $\text{GLU}([A, B]) \in \mathbb{R}^{d}$ has the same dimensionality as the input element. Furthermore, to build deeper convolutional encoder, we add residual connection for each convolutional unit, given by

$$o_i^{l+1} = \text{GLU}(W^l[h_i^{l-1}; \cdots; h_i^{l-1} + b^l]) + h_i^{l-1}.$$ We note that $o_i^{l+1}$ contains information over $k$ input elements and the receptive field of final output can be expanded rapidly by stacking several units. And the output of the two convolutional units of the $l$-th layer is $(o_i^{l+1}, o_2^{l+1}, \cdots, o_n^{l+1})$.

**Attentive Segmentation Unit**

The attentive segmentation unit splits long-form video contents into different segments and leverages question information to learn attentive segment-level representations. Thus, this method can reduce the video sequential length layer by layer to build the hierarchical video structure and also alleviate memory load.

Given the convolution output $(o_1^{l+1}, o_2^{l+1}, \cdots, o_n^{l+1})$, we first devide sequence elements into $n_l$ segments, each containing $H$ elements (i.e. $n_{l-1} = H \times n_l$). We then devise an attention mechanism to aggregate element representations for each segment, where we utilize the question representation as guidance to highlight critical elements. Different from mean-pooling aggregation, this method can effectively filter the irrelevant video contents for subsequent modeling.

Concretely, for the $i$-th segment $(o_i^{l+1}, \cdots, o_{i+H}^{l+1})$ with $H$ elements, we compute the attention weight of $j$-th
element $o_{j}^{H}$ according to the global question representation $h^{Q}$, given by

$$
\alpha_{ij} = w_{s}^{T} \tanh(W_{1}^{T} o_{i-1}^{H,j} + W_{2}^{T} h^{Q} + b_{s}),
$$

where $W_{s}^{1}$, $W_{s}^{2}$ are parameter matrices, $b_{s}$ is the bias vector and the $w_{s}$ is the row vector for computing the attention weight. We then conduct the softmax operation on attention weights and obtain the segment representation $s_{i}^{l}$ by

$$
s_{i}^{l} = \sum_{j=1}^{H} \text{softmax}(\alpha_{ij}) o_{i-1}^{H,j}.
$$

Hence, by the attention segmentation unit, we learn the segment-level video representations $s^{l} = (s_{1}^{l}, s_{2}^{l}, \ldots, s_{n_{l}}^{l})$. This segmentation strategy builds the hierarchical structure of long-form video contents, which is helpful for exploring multi-scale visual clues for question answering. Moreover, this method reduces the video sequential length layer by layer to alleviate the memory load.

### Question-Aware Self-Attention Unit

The question-aware self-attention unit captures the long-range dependencies from long-form video context with question information as guidance, which filters the unnecessary context.

Given the output of the attentive segmentation unit $s^{l} = (s_{1}^{l}, s_{2}^{l}, \ldots, s_{n_{l}}^{l})$ and question semantic representations $h^{l} = (h_{1}^{l}, h_{2}^{l}, \ldots, h_{n_{l}}^{l})$, we can compute the video-to-question attention weights between each pair of sequence element and word, and obtain a video-to-question attention matrix $M \in \mathbb{R}^{n_{l} \times m}$. Specifically, the attention weight of the $i$-th element and $j$-th word is calculated by

$$
M_{ij} = w_{m}^{T} \tanh(W_{m}^{1} s_{i}^{l} + W_{m}^{2} h_{j}^{l} + b_{m}).
$$

We then calculate the self-attention matrix $D$ for video contents based on $M$ as below:

$$
D = M \cdot M^{T}, \quad D \in \mathbb{R}^{n_{l} \times n_{l}}
$$

where each value in $D$ represents the correlation of two video elements. Specifically, each value in $D$ is calculated by

$$
D_{ij} = \sum_{k=1}^{m} S_{ik} S_{kj}^{T}, \quad \text{where } k \text{ represents the index of the } k\text{-th word in the question. That is, we regard the question semantic representations as the middle layer while establishing element-to-element correlation of the video sequence. Compared with the conventional self-attention method, we filter the ineffective video context with question contents as guidance while capturing long-range dependencies.}
$$

We then conduct the softmax operation for each row in the self-attention matrix $D$ and compute self-attention representations for each element, followed by an additive residual connection:

$$
h_{i}^{l} = s_{i}^{l} + \sum_{j=1}^{m} \text{softmax}(D_{ij}) s_{j}^{l}.
$$

Therefore, we obtain the final output of the $l$-th convolutional self-attention layer $h^{l} = (h_{1}^{l}, h_{2}^{l}, \ldots, h_{n_{l}}^{l})$.

### 3.3 Multi-Scale Attentive Decoder Network

In this section, we introduce the multi-scale attentive decoder to consider the multi-scale visual clues from the hierarchical encoder for natural-language answer generation.

Our multi-scale attentive decoder is based on a GRU answer generator. At each time step, we conduct a recurrent operation as follows:

$$
h_{i}^{l} = \text{GRU}(x_{i}, h_{i-1}^{l}),
$$

where $x_{i}$ is the input vector and $h_{i}^{l}$ is the output vector of the $t$-th step. The $x_{i}$ consists of the embedding $w_{i}$ of the input word at the $t$-th step, global question representation $h^{Q}$ and multi-scale video representation $h_{i}^{l}$, given by $x_{i} = [w_{i}; h^{Q}; h_{i}^{l}]$. We then introduce how to develop the multi-scale video representations $h_{i}^{l}$. By hierarchical encoder, the long-form video sequences are exponentially shorten layer by layer, and the output from the top layer of the hierarchical encoder aggregates informative segment-level representations for question answering. However, only considering the top-layer output may ignore fine-grained features in preceding layers and lead to information missing. Thus, after trade-off consideration, we devise a multi-scale attention on encoder outputs of top-K layers. Specifically, for the $l$-layer video representations $h^{l} = (h_{1}^{l}, h_{2}^{l}, \ldots, h_{n_{l}}^{l})$, we compute the attention weight of $i$-th element with the hidden state $h_{i-1}^{l}$ and global question representation $h^{Q}$, given by

$$
\beta_{i} = w_{g}^{T} \tanh(W_{g}^{1} h_{i}^{l} + W_{g}^{2} h_{i-1}^{l} + W_{g}^{3} h^{Q} + b_{g}).
$$

We then conduct the softmax operation on attention weights and obtain the $l$-layer attentive representation $v^{l}$ by

$$
v^{l} = \sum_{i=1}^{n_{l}} \text{softmax}(\beta_{i}) h_{i}^{l}.
$$

Based on top-K attentive representations, we finally get the multi-scale video representation $h_{i}^{l}$ by a mean pooling:

$$
h_{i}^{l} = \frac{1}{K} \sum_{l=L-K+1}^{L} v^{l}.
$$

With $x_{t}$ as input, we get the output vector $h_{o}^{t}$ of GRU networks at the $t$-th step, and further generate the conditional distribution over possible words, given by

$$
p_{t} = \text{softmax}(W_{o} h_{o}^{t} + b_{o}, \in \mathbb{R}^{T}),
$$

where $\tilde{a}_{i}$ is the $t$-th generated word and $T$ is the size of the vocabulary. Finally, we apply the maximum likelihood estimation to train our encoder-decoder networks in an end-to-end manner, given by

$$
\mathcal{L}_{ML} = - \sum_{t=1}^{T} \log p_{t}(\tilde{a}_{i}|\tilde{a}_{1:t-1}).
$$

### 4 Experiments

#### 4.1 Dataset

We conduct experiments on an open-ended long-form video question answering dataset [Zhao et al., 2018], which is constructed from the ActivityCaption dataset [Krishna et al., 2017] with natural-language descriptions. The average video time in the dataset is approximately 180 seconds and the longest video even lasts over 10 minutes. The question-answer pairs contain five types corresponding to the object, number, color, location and action for the video contents. The first four types of questions mainly focus on the appearance visual features in videos but motion features are necessary to the action related question answering. The details of this dataset are summarized in Table 1.
4.2 Implementation Details

In this section, we introduce data preprocessing and model settings.

We first resize each frame to $112 \times 112$ and then employ the pre-trained 3D-ConvNet [Tran et al., 2015] to extract the 4,096-d feature for each unit, which contains 16 frames and overlaps 8 frames with adjacent units. We then reduce the dimensionality of the feature from 4,096 to 500 using PCA. Next, we set the maximum sequence length to 512 and down-sample overlong sequences to this length. As for the question, we employ the pre-trained word2vec model [Mikolov et al., 2013] to extract the semantic features word by word. The dimension of word features is 300 and the vocabulary size $T$ is 10000.

In our HCSA, we set the layer number $L$ of the hierarchical convolutional self-attention encoder to 3. And the segmentation factor $H$ in the attentive segmentation unit is set to 4. To avoid heavy computational cost, we only consider top-2 layers ($K = 2$) of the hierarchical encoder for the multi-scale attentive decoder. Moreover, we set convolution kernel width $k$ to 5, convolution dimension to 256 and the dimension of the hidden state of GRU networks to 256 (512 for BiGRU while question encoding). And the dimensions of the linear matrix in all kinds of attention are set to 256. During training, we adopt an adam optimizer to minimize the loss and the learning rate is set to 0.001.

4.3 Performance Criteria

We evaluate the performance of open-ended video question answering based on evaluation criteria BLEU-1, WUPS@0.0 and WUPS@0.9. Since the length of answers is relatively short, we mainly focus on word-level evaluation. The BLEU-1 is for accurate word matching and WUPS [Malinowski and Fritz, 2014] accounts for word-level ambiguities. Given the ground-truth answer $a = \{a_1, a_2, \ldots, a_r\}$ and the generated answer $\hat{a} = \{\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_r\}$, the WUPS is given by

$$WUPS = \frac{1}{|Q|} \sum_{q \in Q} \min \{ \frac{1}{r} \sum_{a_i \in a} \max_{\hat{a}_j \in \hat{a}} WUP_q(a_i, \hat{a}_j), \frac{1}{r} \sum_{\hat{a}_j \in \hat{a}} \max_{a_i \in a} WUP_q(\hat{a}_j, a_j) \},$$

where the $WUP_q(\cdot)$ [Wu and Palmer, 1994] calculates word similarity by the WordNet [Fellbaum, 1998], given by

$$WUP_q(a_i, \hat{a}_j) = \begin{cases} WUP(a_i, \hat{a}_j), & WUP(a_i, \hat{a}_j) \geq \gamma \\ 0.1 \cdot WUP(a_i, \hat{a}_j), & WUP(a_i, \hat{a}_j) < \gamma \end{cases}$$

We set the threshold $\gamma$ to 0 and 0.9 and denote the two criteria by WUPS@0.0 and WUPS@0.9, respectively.

4.4 Performance Comparisons

The open-ended video question answering is an emerging task, thus we compare our HCSA method with existing open-ended methods and meanwhile extend some conventional multi-choice methods for performance evaluation. Specifically, we add a GRU recurrent answer generator with attention mechanism to the end of those non-open-ended models.

- **MN+** method [Zeng et al., 2017] is the extension of end-to-end memory network algorithm, where we add a bi-LSTM network to encode the sequence of video frames.
- **STVQA+** method [Jang et al., 2017] utilizes the spatial and temporal attention strategies on videos to answer related questions.
- **CDMN+** method [Gao et al., 2018] proposes a motion-appearance co-memory network to simultaneously learn the motion and appearance features.
- **UNIFY** method [Xue et al., 2017] applies the sequential video attention and temporal question attention for open-ended video question answering.
- **AHN** method [Zhao et al., 2018] divides entire videos into several segments and adopts a hierarchical attention to model video presentations for answer generation.

The former three approaches are originally developed for multi-choice video question answering and we extended them into the open-ended form. Table 2 shows the overall experimental results of all methods on three criteria BLEU-1, WUPS@0.9 and WUPS@0.0. Table 4 and 5 demonstrate the evaluation results of different question types on BLEU-1 and WUPS@0.9, respectively. Moreover, we adjust the parameter number in different methods at the same magnitude for fairly evaluating the time consumption, shown in Table 3.

The training and inference time for each epoch only contain the network execution time. The experiment results reveal some interesting points:

- The methods based on attention mechanism, UNIFY, STVQA+, CDMN+, AHN, and HCSA achieve better...
Table 4: Results on BLEU-1 with different question types.

<table>
<thead>
<tr>
<th>Method</th>
<th>Object Number</th>
<th>Color Location</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>MN+</td>
<td>25.02</td>
<td>59.58</td>
<td>22.53</td>
</tr>
<tr>
<td>UNIFY</td>
<td>31.01</td>
<td>74.98</td>
<td>26.21</td>
</tr>
<tr>
<td>STVQA+</td>
<td>30.53</td>
<td>75.56</td>
<td>26.62</td>
</tr>
<tr>
<td>CDMN+</td>
<td>31.06</td>
<td>76.43</td>
<td>24.20</td>
</tr>
<tr>
<td>AHN</td>
<td>31.48</td>
<td>78.72</td>
<td>24.63</td>
</tr>
<tr>
<td>HCSA</td>
<td>34.48</td>
<td>79.49</td>
<td>26.91</td>
</tr>
</tbody>
</table>

Table 5: Results on WUPS@0.9 with different question types.

<table>
<thead>
<tr>
<th>Method</th>
<th>WUPS@0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>MN+</td>
<td>29.22</td>
</tr>
<tr>
<td>UNIFY</td>
<td>38.63</td>
</tr>
<tr>
<td>STVQA+</td>
<td>38.75</td>
</tr>
<tr>
<td>CDMN+</td>
<td>38.79</td>
</tr>
<tr>
<td>AHN</td>
<td>39.43</td>
</tr>
<tr>
<td>HCSA</td>
<td>41.17</td>
</tr>
</tbody>
</table>

4.5 Ablation Study and Visualization

To prove the contribution of each component of our HCSA method, we next conduct some ablation studies. Concretely, we discard or change one component at a time to generate an ablation model. We first replace the segment attention in the attentive segmentation unit with a conventional mean-pooling, denoted by ASU(MP), and further remove attentive segmentation unit as w/o. ASU. We then replace the question-aware self-attention with a typical self-attention, denoted by QSU(SA), and thoroughly discard the question-aware self-attention unit as w/o. QSU. We finally denote the model that only considers the top-layer video representations using a thermodynamic diagram, where the darker color represents the higher correlation. For each step of answer generation, the multi-scale attention produces a weight distribution over the video semantic representations of top-2 layers. We note that our proposed method can attend the semantically related visual contents and ignore these irrelevant features at each step. It suggests our proposed decoder effectively incorporates the multi-scale visual clues for high-quality open-ended long-form video question answering.

5 Conclusion

In this paper, we propose a fast hierarchical convolutional self-attention encoder-decoder network for open-ended long-form video question answering. We first propose a hierarchical convolutional self-attention encoder to efficiently model long-form video contents, which builds the hierarchical structure for video sequences and captures question-aware long-range dependencies from video content. We then devise a multi-scale attentive decoder to incorporate multi-layer video representations for natural-language answer generation. The extensive experiments show the effectiveness and efficiency of our method.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No.61602405, No.U1611461, No.61751209 and No.61836002, Joint Research Program of ZJU and Hikvision Research Institute. This work is also supported by Alibaba Innovative Research and Microsoft Research Asia.
References


