Equivariant Adaptation of Large Pretrained Models

Abstract

This paper explores the adaptation of video alignment to improve multi-step inference. Specifically, we first utilize VideoCLIP to generate video-script alignment features. Afterwards, we ground the question-relevant content in instructional videos. Then, we reweight the multimodal context to emphasize prominent features. Finally, we adopt GRU to conduct multi-step inference. Through comprehensive experiments, we demonstrate the effectiveness and superiority of our method.

1 Introduction

This paper addresses the critical task of assisting users in navigating unfamiliar events for specific devices by providing step-by-step guidance using knowledge acquired from instructional videos. Due to the substantial disparity among specific tasks, the integration of multimodal input, and the complexity of multi-step inference, this is still a challenging task.

Several studies have been proposed to address this task. For instance, one study proposes a Question-to-Actions (Q2A) Model, which employs vision transformer (ViT) and BERT to extract visual and textual features, respectively. Moreover, attention mechanisms are leveraged to anchor question-relevant information in instructional videos. Another study proposes a two-stage Function-centric approach, which segments both the script and video into function clips instead of sentences or frames. Additionally, they substitute BERT with XL-Net for text encoding. Despite the advancements achieved through these techniques, all of them adopt the unaligned pretrained encoders to extract visual and textual features, leading to significant semantic gaps between modalities, thereby hindering better results.

To alleviate the negative effects of modalities unalignment, in this paper, we leverage pretrained video-text models to achieve instructional video-text alignment, facilitating a more robust grounding of question-relevant knowledge for multi-step inference. We build the pipeline with four steps: Instructional Video Alignment, Question-Aware Grounding, Multimodal Context Reweighting and Multi-Step Inference. Specifically, we employ pretrained VideoCLIP for generating video-script alignment features, which are beneficial to cross-modal grounding. Subsequently, we anchor the question-relevant content in instructional videos by the combination of hard and soft grounding. Afterwards, we leverage additive attention to adjust the weighting of the multimodal context to emphasize the salient features. Finally, we employ GRU for performing multi-step inference. We reduce the proportion of teacher forcing linearly to bridge the gap between training and inference, which boosts the multi-step inference.

2 Problem Definition

In this section, we formulate the problem of AQTC.

Given an instructional video, which contains numerous frames and scripts, AI assistant extracts relevant information from the video in accordance with the user2̌019s question q. Then, it deduces the correct answer ai j based on the image U as perceived by the user, from the candidate answer set Ansi = ai 1, ai 2, ..., ai n in i-th step. Following previous work, we segment the video into several clips based on scripts. Each clip illustrates one specific function of the device in video. We concatenate

.

these clips to form the visual function sequence as [F v 2 , ..., F v 1 , F v m] and the textual function sequence as [F t 1, F t 2, ..., F t m], where F v i comprises all frames of the i-th function2019s clip, and F t i contains all script sentences of the i-th function2019s clip. To adapt AI assistant to the user2019s view, following previous work, we mask the referenced button related to candidate answers in user images U , denoted as bk.

3 Method

In this section, we will introduce the details of our method. Our method consists of four steps: Instructional Video Alignment, Question-Aware Grounding, Multimodal Context Reweighting and Multi-Step Inference.

3.1 Instructional Video Alignment

To align the videos and the text for better cross-modal understanding, we leverage pretrained VideoCLIP to generate the features of instructional videos. For the video part, we initially utilize pretrained S3D to generate an embedding for each second of the video, with a frame rate of 30 frames per second. Next, to represent each function within the videos, we utilize the pretrained visual transformer from VideoCLIP to process the embeddings generated by S3D in each function. Then, we apply average pooling over the processed sequence of embeddings to form the video embedding Vi corresponding to a given visual function F v i . For the text part, we use the pretrained textual transformer of VideoCLIP to encode the scripts of a textual function F t i . Similarly, we employ average pooling to aggregate the processed sequence of text, generating the text embedding Ti of a given textual function F t i . Finally, we obtain the video feature sequence [V1, V2, ..., Vm] and the text feature sequence [T1, T2, ..., Tm] of the given function sequence.

Besides, we also utilize VideoCLIP to encode the questions q, the answer ai j and the masked button image bk. We duplicate the images 30 times to ensure consistent video encoding. We get the question feature Q, answer feature Ai j and visual button feature Bk.

3.2 Question-Aware Grounding

Owing to the extensive pretraining of VideoCLIP on a vast collection of videos, the features of videos and text are cross-modal aligned. Therefore, we can utilize the question Q to ground the video and text feature sequence directly. Specifically, we leverage three grounding mechanisms: soft, hard and combined grounding. Soft grounding employs attention to learn the similarity between the question feature Q and the video feature sequence [V1, V2, ..., Vm] directly. And, it uses another attention network to compute the similarity between the question feature Q and the text feature sequence [T1, T2, ..., Tm]. Soft grounding adopts the similarity from two attention networks to perform a weighted average of the two feature sequences, respectively. Instead of relying on deep learning methods, hard grounding follows previous work, which uses TF-IDF model to calculate the similarity between the question q and each textual function F t i from textual function sequence [F t 1, F t 2, ..., F t m]. Then, it uses the similarities as the weights to compute the averages of the video feature sequence [V1, V2, ..., Vm] and the text feature sequence [T1, T2, ..., Tm], respectively. Besides, the combined grounding utilizes soft grounding and hard grounding simultaneously. Then, the two features from two grounding methods are averaged. Ultimately, we obtain the aggregated question-aware video feature V and text feature T .

3.3 Multimodal Context Reweighting

After obtaining multimodal question-aware context features from instructional videos, we need to model the answers to determine the correct one. Specifically, we utilize the gate network to fuse the candidate answer feature Ai j with the corresponding button feature Bk, which generates the multimodal answer feature ŏ2c6Ai j. We concatenate these multimodal contexts into a sequence [V, T, Q, ŏ2c6Ai j] for each candidate answer. Due to the varying importance of different context features in determining the correct answers, we utilize additive attention to reweight the multimodal context and get the fused feature. Finally, the fused feature is processed using a two-layer MLP to obtain the candidate answer context feature C i j.

3.4 Multi-Step Inference

Owing to the requirement for multi-step guidance in order to respond to the given questions, it is essential for models to perform multi-step inference. Following previous work, we utilize GRU to infer the current correct answer by incorporating historical knowledge. Specifically, we feed the previous hidden state H i2̌2121 and the contextual features C i j of candidate answers in Ansi into the GRU. Then, the resulting current hidden state H i j for each candidate answer in Ansi is utilized to predict the correct answer in the i-th step. We adopt a two-layer MLP and the softmax function on the concatenated current hidden states [H i 1, H i 2, ..., H i n] to generate the probability of the correct answer. Cross entropy is used to compute the loss. While previous works utilize the state of the ground truth as the historical state of the next step H i. This causes a huge gap between training and inference. To bridge this gap, we reduce the reliance on teacher forcing linearly. In other words, we choose the hidden state of the most probable answer predicted by models as the historical state of the next step H i, when a sample is selected for autoregressive training.

4 Experiments

4.1 Dataset and Implementation Details

We use AssistQ train@22 and test@22 sets to train and validate. And we test our model on the AssistQ test@23 dataset.

In our experiments, we use Adam optimizer with a learning rate 10̌2124. The batch size is set to 16, the maximum training epoch is 100, and we adopt early stopping. We randomly select 5

4.2 Performance Evaluation

We present the performance evaluation on the test dataset in Table 1a. We find that our method outperforms baseline methods. This superiority can be attributed to our utilization of a video-text aligned pretrained encoder for feature extraction. The aligned features are beneficial to multi-step inference. Furthermore, our method exhibits improved performance when the results are ensembled.

Table 1: Performance evaluation and impact of pretrain features.

Methods	R@1 (%)	R@3 (%)
Q2A	67.5	89.2
Question2Function	62.6	87.5
Ours	75.4	91.8
Ours (Ensemble)	78.4	93.8

Methods	R@1 (%)	R@3 (%)
ViT+XL-Net	63.9	86.6
VideoCLIP (Ours)	75.4	91.8

Table 2: (b) Impact of pretrain features.

4.3 Ablation Study

Pretrain Feature To validate the efficacy of video-text aligned features, we conduct the ablation study, which adopts ViT for processing the visual features and XL-Net for processing the text features. As shown in Table 1b, we observe that the performance of method that uses the unaligned features drops sharply.

Grounding Methods To validate the effectiveness of various grounding methods, we use different grounding techniques to train this model. The result is presented in Table 2. We find that the model achieves optimal performance when the text grounding leverages combined grounding and the video grounding utilizes soft grounding.

Text Grounding	Video Grounding	R@1 (%)	R@3 (%)
Soft	Soft	75.4	91.8
Hard	Soft	75.1	89.2
Soft	Hard	73.8	90.5
Hard	Hard	71.8	89.8

Table 3: Impact of grounding methods.

Methods	R@1 (%)	R@3 (%)
Ours	75.4	91.8
w/o reweighting	72.1	89.5
w/o SSL	72.5	92.1

Table 4: (a) Impact of the reweighting mechanism and SSL.

Reweighting Mechanism We show the result of the model without attention reweighting in Table 3a. We observe a considerable decrease in performance for the model lacking attention reweighting. This is because the attention reweighting can discern and prioritize the most informative features within complex multimodal contexts.

Multi-Step Inference We evaluate different multi-step inference strategies, as demonstrated in Table 3b. We find that the performance of TeacherForcing is inferior to that of the Linear Decay strategy, which is employed by our approach. This is because TeacherForcing widens the gap between training and inference. We also observe that Linear Decay outperforms AutoRegression. This is because teacher forcing is beneficial in preventing models from accumulating mistakes during the early stages of training.

SSL The performance of the w/o SSL model exhibits a significant drop, as shown in Table 3a.

5 Conclusion

In this paper, we present a solution aimed at enhancing video alignment to achieve more effective multi-step inference for the AQTC challenge. We leverage VideoCLIP to generate alignment features between videos and scripts. Subsequently, we identify and highlight question-relevant content within instructional videos. To further improve the overall context, we assign weights to emphasize prominent features. Lastly, we employ GRU for conducting multi-step inference. Besides, we conduct exhaustive experiments to validate the effectiveness of our method.

Methods	R@1 (%)	R@3 (%)
Linear Decay (Ours)	75.4	91.8
AutoRegression	74.4	91.1
TeacherForcing	74.1	88.5

Table 5: (b) Impact of multi-step inference strategies.