Video Question Answering via Gradually Refined Attention over Appearance and Motion*

Dejing Xu‡, Zhou Zhao‡, Jun Xiao‡, Fei Wu‡, Hanwang Zhang§, Xiangnan He¶, Yueting Zhuang‡

‡Zhejiang University, §Columbia University, ¶National University of Singapore
{xudejing, zhaozhou, yzhuang}@zju.edu.cn, {junx,wufei}@cs.zju.edu.cn
{hanwangzhang, xiangnanhe}@gmail.com

ABSTRACT

Recently image question answering (ImageQA) has gained lots of attention in the research community. However, as its natural extension, video question answering (VideoQA) is less explored. Although both tasks look similar, VideoQA is more challenging mainly because of the complexity and diversity of videos. As such, simply extending the ImageQA methods to videos is insufficient and suboptimal. Particularly, working with the video needs to model its inherent temporal structure and analyze the diverse information it contains. In this paper, we consider exploiting the appearance and motion information resided in the video with a novel attention mechanism. More specifically, we propose an end-to-end model which gradually refines its attention over the appearance and motion features of the video using the question as guidance. The question is processed word by word until the model generates the final optimized attention. The weighted representation of the video, as well as other contextual information, are used to generate the answer. Extensive experiments show the advantages of our model compared to other baseline models. We also demonstrate the effectiveness of our model by analyzing the refined attention weights during the question answering procedure.

KEYWORDS

Video Question Answering; Attention Mechanism; Neural Network

1 INTRODUCTION

Obtaining information from videos is important and valuable for numerous applications especially when a vast number of videos are produced nowadays. It is nearly impossible for us to look these videos through. Video question answering (VideoQA) can help us quickly acquire the necessary information we seek from videos, which is beneficial to various real-life applications [4, 8, 28, 29, 36].

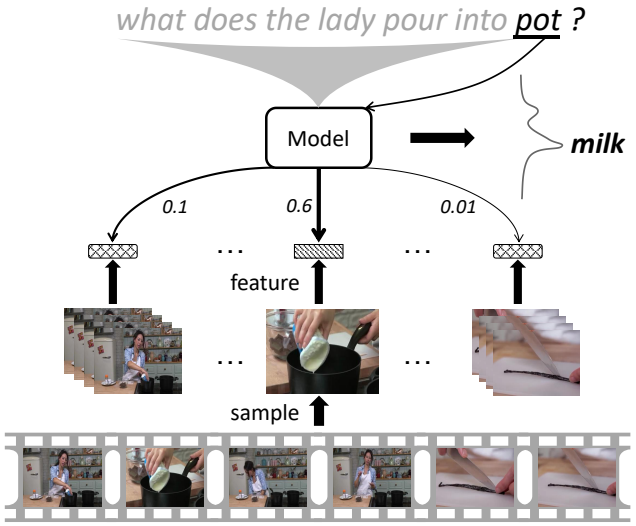

Figure 1: Given the video and the question, our model first samples the video in frame-level and clip-level, then extracts appearance and motion features, while the question is processed in word-level to refine the attention gradually. The numeric values beside the lines indicate the refined attention after the last word being processed.

Compared to the understanding of images [5, 27, 44, 45], the understanding of videos is more difficult. Videos not only contain one more dimension of time, but also include extra information channels such as the audio and text in most cases. To measure the understanding ability of models on videos, different intermediate tasks are proposed such as video classification [12, 19, 35, 41] and action recognition [15, 20, 22, 24, 46]. Recently video captioning has emerged where models need to describe the most informative content of video clips using natural language sentences. The task gains lots of interest in the research community and several works have shown promising results [17, 25, 40]. Though describing the video using natural language is close to how humans perceive it, existing models tend to describe the video in a short sentence, which may miss lots of important details. Many metrics such as BLEU, METEOR, ROUGE, and CIDEr have been proposed for the evaluation of sentences, but it is still hard to evaluate due to the ambiguity and variety of natural language as well as the differences among subjective assessments [43].

*The corresponding author is Jun Xiao.

MM '17, October 23–27, 2017, Mountain View, CA,USA

ACM ISBN 978-1-4503-4906-2/17/10…$15.00
https://doi.org/10.1145/3123266.3123427

VideoQA is proposed recently, which follows the pattern of image question answering (ImageQA) [2] except for the media type. Given a video and a question, the task requires the model to output an answer. Because videos and questions are arbitrary, the model succeeding in VideoQA has the ability to analyze the unconstrained videos and questions, thus indicating a better understanding of videos. As have mentioned above, the model in video captioning tends to describe the video in general sentences with details missing; in VideoQA, questions may involve diverse details of the video, and answers cannot be so general. To give the answer, a model needs to analyze the question carefully and focuses on the important part of the video. Besides, since the details are always simple and concrete, the answer of the question is always much shorter than the description of the video, which makes the evaluation easier. The above properties also make VideoQA a better choice to measure the progress in video understanding. A system that succeeds in ImageQA typically needs a more detailed understanding of the image and complex reasoning than a system producing generic image captions [2], which is the same when it refers to videos.

Currently, existing works on VideoQA are sparse as the task is relatively new. [43] presents four models to solve the VideoQA task. All of these models are extended from models for other tasks such as video captioning and ImageQA. Though these models are proved powerful in their dedicated task, simply extending them is inappropriate since they may weaken or ignore the temporal information of videos which is the most distinctive feature compared to static images. Besides, videos contain rich data in multiple channels, and the simply extended models cannot leverage them well. Another weakness is that these models always encode the whole question as one single feature, which may be not expressive enough to reveal the information held by the question. Moreover, some words in the question are significant to give the correct answer, and these words may also be buried in such coarse-grained feature.

In this paper, we propose an end-to-end model for VideoQA. The model first samples the video as a series of frames and clips, from which the appearance and motion features are extracted. Then the model reads the question word by word and refines its attention over these features with the interaction between the frame-level and clip-level. When all words of the question are processed, the model generates the final optimized attention which can be used to fuse the appearance and motion features as the representation of the video. Both the coarse-grained question feature and fine-grained word feature are used in the procedure. Taking what Figure 1 shows as an example, the model reads the question *"what does the lady pour into pot?"* sequentially and finally finds that the frame presented in the middle of sample results is the most relevant and important to give the answer *"milk"*.

To summarize, the main contributions of the paper are as follows:

- We propose to solve the VideoQA task by utilizing the appearance and motion information resided in videos.
- Our proposed model refines the attention of the video using both coarse-grained question feature and fine-grained word feature together as guidance.
- We generate two datasets and evaluate our proposed model. The extensive experiments show that our model achieves promising results.

The rest of the paper is organized as follows. We first introduce a few related works in Section 2. The details of our proposed model are explained in Section 3. In Section 4, we describe the settings of experiments, present several results, and analyze the results in depth. Finally, we conclude our paper in Section 5.

2 RELATED WORK

In this section, we briefly review some works related to video question answering (VideoQA) and inspires the design of our model. Since VideoQA is a fairly new task, there are only a few works to refer. Here we also consider two related tasks which are video captioning and image question answering (ImageQA).

2.1 Video Captioning

Video captioning aims at generating sentences which describe the content of the video. There exist lots of works targeting on this task. [26] samples several frames from the video and extracts the fc_7 layer's activations of the convolutional neural network (CNN) as the feature for each frame. After the feature extraction, the model mean pools all of these features across the entire video and inputs the pooled features to a two layer Long Short-Term Memory (LSTM) network. The LSTM network outputs one word at each timestep, based on the video features and the previous word until it outputs the end-of-sentence tag. [42] proposed to attend on top of spatiotemporal object proposals in the video, integrate it with state-of-the-art image classifiers, object detectors, high-level semantic features (SVO) and use the recurrent neural network (RNN) to generate the caption of the video.[38] proposed a novel 3D CNN-RNN encoder-decoder architecture which captures local spatiotemporal information by using the 3D CNN feature. Instead of mean pooling the video features, the method attends on each feature when generating the next word of the sentence. [6] proposed a model with memory-augmented attention which enhances the attention mechanism. The model utilizes memories of past attentions performed over the video when thinking about where to attend to in current timestep.

2.2 Image Question Answering

Given an image and a natural language question, image question answering (ImageQA) requires the model to provide a correct answer. [37] proposed a model with stacked spatial attention over different regions of the image. The question is processed by CNN or LSTM network to extract a semantic vector as its feature. The image is processed by VGG network [23] and the activations in the last pooling layer are extracted as features for image regions. The model first uses the question feature to query the image region features in the first visual attention layer, then combines the question feature and the retrieved image feature to form a refined query vector, and finally query the image vectors again in the second attention layer. Since some questions can not be answered using the image only, [31] brings external knowledge into the model. Besides the image and the question, the model also retrieves some facts from the external knowledge base. Vector representations of region proposals, image captions together with the retrieved knowledge are used to form a fused representation, then the representation is fed to a LSTM network which reads the question and generates an

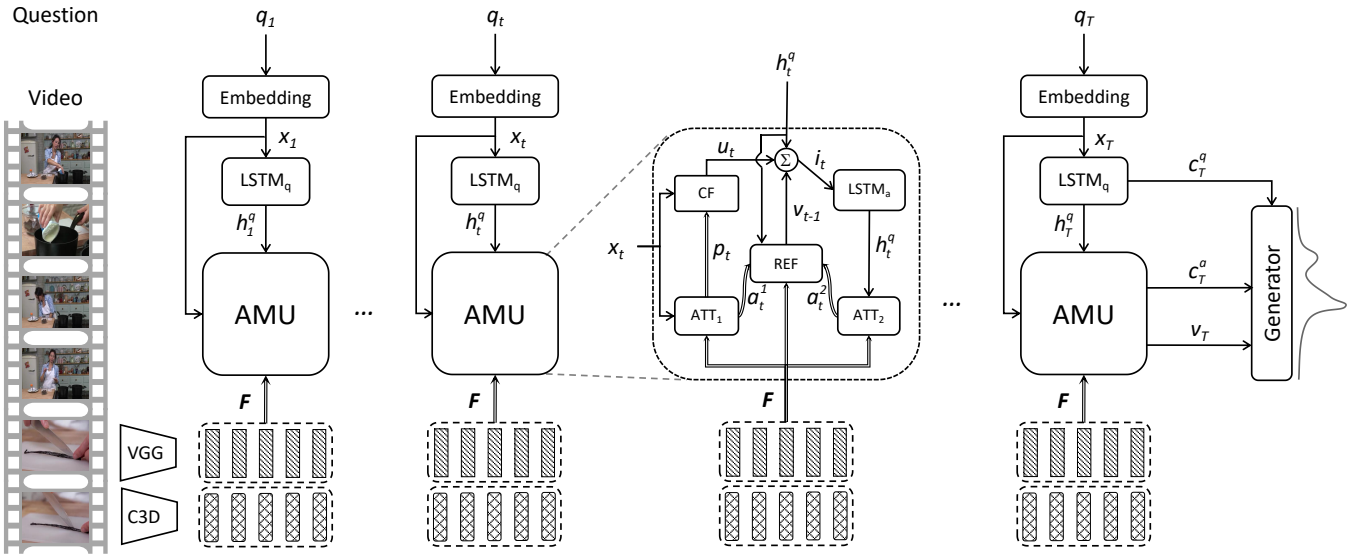

Figure 2: Our model transforms the words by the embedding layer and manipulates the attention in Attention Memory Unit (AMU). The model processes the question word by word while AMU generates and refines the attention over appearance and motion features of the video at each timestep. After all words are processed, the final attention is used to fuse both features as the representation of the video. Together with other contexts, our model outputs the answer. The double line in the figure indicates features in two channels.

answer. Dynamic memory network [14] obtained high accuracy on many language tasks. [32] improved the input module of dynamic memory network to consume image regions as [37] and made the model capable of performing question answering on images. The model also uses a bidirectional Gated Recurrent Unit (GRU) network to update the region features, which make the information flow among regions in the image. [33] proposed a model which lets each word of the question select a related image region, then gathers the weighted representation to generate the answer.

2.3 Video Question Answering

Video question answering (VideoQA) is a relatively new task, where a video and a natural language question are provided and the model needs to give the right answer. [51] presents an encoder-decoder approach to answer the multiple-choice questions of the video, but the form of the question is "fill-in-the-blank" which is different from open-ended question answering proposed here. [43] is the first work in VideoQA to our knowledge. The work simply extends several existing models from other tasks such as video captioning and ImageQA. All of the extended models extract the feature of the question by using the LSTM network. The extracted feature is coarse and lacks the ability to reveal the information held by key words since the single semantic vector may conceal the feature of specific words. Furthermore, the appearance and motion features of the video are extracted but only fused by naive mean pooling. We think the two channels of features should be exploited with interaction more effectively. Recent papers [39, 49, 50] also employ the attention mechanism over videos. Unlike these studies, our paper proposes to refine the attention gradually using both coarse-grained question feature and fine-grained word feature.

3 METHODS

In this section, we first define several necessary annotations, then present the framework of our proposed model in details.

We currently focus on videos which are short in time duration and have only a few scenes. Given the video V and the question Q, the goal of the video question answering (VideoQA) is to give the appropriate answer A. Since the video always contains lots of frames per second which are redundant, it is sampled in evenly distributed frames and clips which can be thought as a compressed representation of the whole video. The clip consists of 16 consecutive frames which hold the basic motion information. The number of frames are chosen to be compatible with the feature extractor. After the sampling process, the video V is represented as a series of frames and clips. The simply compressed representation not only reduces the redundant frames which may increase the computation burden of the model but also keeps the information of the video in all timesteps as much as possible. The question Q is represented as a series of words respectively. For the same video, there may be lots of different questions, thus the model needs to realize what the question asks and find the necessary information from the video. The length of answer A is not very long since what the question asks is always specific and can be answered in a few words. The concise answer also makes the evaluation easier, which is important for comparing different models and measuring the progress in the task.

To overcome the weakness of the models proposed in [43], we propose an end-to-end model for VideoQA. Our model utilizes the appearance and motion information in the video and analyzes the question more carefully. As Figure 2 shows, for a given video, the appearance and motion features from the video are extracted first,

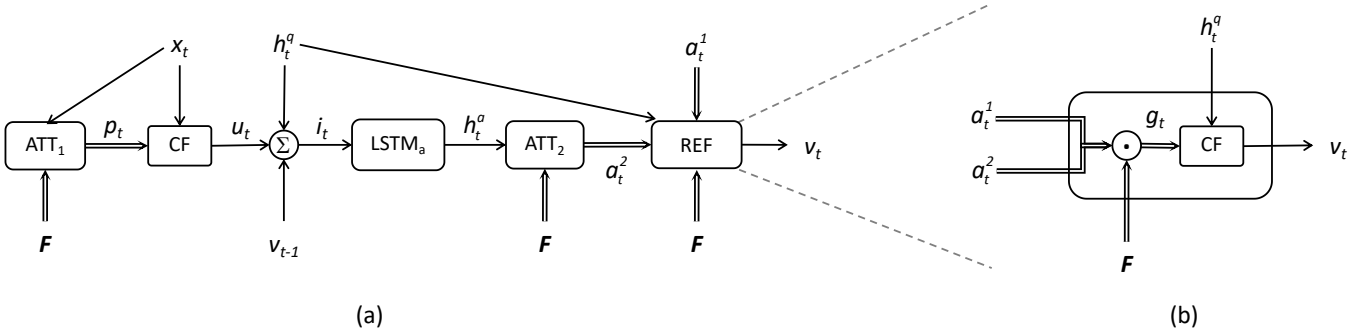

Figure 3: The operation blocks in AMU is unrolled based on its execution order from left to right in (a), and the details of operation REF is presented in (b). The double line in the figure indicates features in two channels.

then the question is analyzed word by word and the attention over these features are refined by Attention Memory Unit (AMU) at each timestep. After the final word of question is processed, the model generates the refined attention for the video which is most relevant and valuable to answer the specific question. The model uses this attention to fuse appearance and motion features and get the representation of the video. To generate the answer, other contexts such as question information and attention history are also used for reference. The details of the model are explained as Feature Extraction, Attention Memory Unit, and Answer Generation in the following.

3.1 Feature Extraction

Feature extraction has been studied in many research fields [47, 48]. The frames of the video are static images containing different objects, which occupy a large portion of information in the video. We can extract appearance features from frames. Besides static objects, motion is another information channel contained in videos that makes difference between videos and images. The motion features can be extracted from the clips of the video which are composed of 16 consecutive frames. We extract features in frame-level and clip-level to get a series of vector representations of the video. There exist other features we can extract from the video such as audio features and text features. Currently, we only consider the appearance and motion channels since they are available in all videos and enough to explain our model. For questions, we use the embedding technique which is explained below.

Appearance. Recent works [4, 11] show that the activations in the deeper layer of CNN can generalize well to other tasks, which means the activations can represent the semantics of the image. A lot of related tasks such as object detection and localization are also tackled with this kind of feature [7, 21]. In our model, we select to use VGG network [23] as the frame-level appearance feature extractor because it is widely used and shows promising results in the literature. For a given video, we represent its appearance features as $\boldsymbol{F}_a = [\boldsymbol{f}_1^a, \boldsymbol{f}_2^a, \ldots, \boldsymbol{f}_N^a]$,where N is number of frames sampled in the video and superscript a indicates appearance.

Motion. [10] proposes the C3D network to perform action recognition and shows promising results in several datasets. The activations in the deeper layer of C3D network are also used in lots of tasks related to videos [38, 42, 43] and show its ability to capture the dynamic information of videos. With the same reason as before, we use C3D network as the clip-level motion feature extractor. For a given video, its sampled clips are processed by C3D network and the extracted motion features are represented as $\boldsymbol{F}_m = [\boldsymbol{f}_1^m, \boldsymbol{f}_2^m, \ldots, \boldsymbol{f}_N^m]$, where N is the number of clips sampled in the video and superscript m indicates motion.

Question. Question can be represented as a series of words annotated as $\boldsymbol{Q} = [q_1, q_2, \ldots, q_T]$. There are many ways to encode words such as one-hot encoding and bag-of-words encoding. Recent works [16, 18] represent a word by a fixed-length continuous vector which is called word embedding. The representation is compact and can capture the high-level semantic meaning of the word. In our model, we also use the embedding layer to transform the word q_t to its semantic embedding $\boldsymbol{x}_t$.

After the features of the video and question are extracted, a novel attention mechanism is applied to generate the attention over the video features based on the question.

3.2 Attention Memory Unit

In this section, we present the attention process which is the core of our model. We first present the attention process in a global perspective, then we explain operation blocks inside Attention Memory Unit (AMU) in details.

The words of the question are processed sequentially and a novel attention mechanism is applied along the procedure. As we can see in Figure 2, the model first uses an embedding layer to transform the input word to its embedding $\boldsymbol{x}_t$, which holds the semantic information of the current word. The word embedding $\boldsymbol{x}_t$ is then fed into the LSTM$_q$, the hidden state of which is considered to remember the information about the processed question part. Both the word embedding $\boldsymbol{x}_t$ and the hidden state $\boldsymbol{h}_t^q$ of LSTM$_q$ are inputted to AMU to generate and refine the attention over the appearance and motion features.

As shown in Figure 2, AMU takes the current word embedding, question information, and video features as inputs, then performs several steps to refine the attention over video features. For clarity, we use the double line to represent the features that contain two channels. There are four main operation blocks in AMU which are Attention (ATT), Channel Fusion (CF), Memory (LSTM$_a$) and Refine (REF). Together with several transformation operations, they

constitute the gradually refined attention mechanism of our proposed model. We first give an overview of the attention mechanism performed in AMU, then show the concrete details of involved operations.

In Figure 3, we unroll the operation blocks inside AMU according to its execution order for clarity. ATT_1 performs the initial attention over $\boldsymbol{F}$ based on current word embedding $\boldsymbol{x}_t$ and attends on video features related to current word. The weighted sum of appearance features $\boldsymbol{p}_t^a$ and motion features $\boldsymbol{p}_t^m$ are then fused by CF, which assigns importance score for each channel, and get the intermediate fused representation of video $\boldsymbol{u}_t$. The hidden state $\boldsymbol{h}_t^q$ of LSTM_q, the previously generated video representation $\boldsymbol{v}_{t-1}$ and the intermediate video presentation $\boldsymbol{u}_t$ are added together to form the input of LSTM_a, which remembers all performed attention operations. ATT_2 uses $\boldsymbol{h}_t^a$ to perform the second attention over $\boldsymbol{F}$. The first attention weights $\boldsymbol{a}_t^1$ and second attention weights $\boldsymbol{a}_t^2$ are refined in REF and the video representation $\boldsymbol{v}_t$ is generated which will be used in next timestep. In the following, we will explain the each operation blocks concretely.

Attention. Given a question about the video, only a subset of the frames or clips are related in most of the time. It is these features which are more useful to give the answer. The attention mechanism aims to assign weights to appearance and motion features of the video separately and attends to useful features by weighted combination of them. There are two attention operations ATT_1 and ATT_2 in AMU. We take ATT_1 as an example to explain the formulation of the attention operation. As we can see from Figure 2, ATT_1 uses the word embedding $\boldsymbol{x}_t$ to perform the attention over video features $\boldsymbol{F}$. Here we omit the notation of appearance or motion for simplicity. The operation is performed on each channel of the features. The attention mechanism can be formulated as follows:

$$\begin{aligned} e_i &= \tanh(\boldsymbol{W}_f \boldsymbol{f}_i + \boldsymbol{b}_f)^T \tanh(\boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{b}_x) \\ a_i &= \frac{\exp(e_i)}{\sum_{i=1}^{N} \exp(e_i)} \end{aligned} \tag{1}$$

The weights a_i reflects the relevance between the current word and the ith feature, $\boldsymbol{W}_f$ and $\boldsymbol{W}_x$ are used to transform both word embeddings and video features to the same underlying embedding space. With the attention weight a_i, the fused feature $\boldsymbol{p}_t$ is computed as follows:

$$\boldsymbol{p}_t = \sum_{i=1}^{N} a_i \tanh(\boldsymbol{W}_f \boldsymbol{f}_i + \boldsymbol{b}_f) \tag{2}$$

The $\boldsymbol{p}_t$ is the representation of the video attended by current word of the question. The ATT_1 here enhances the influence of current word when answering the question. Later ATT_2 will use $\boldsymbol{h}_t^a$ to perform another attention operation which also generate second attention weights.

Channel Fusion. After getting the features $\boldsymbol{p}_t$, which is actually composed of $\boldsymbol{p}_t^a$ and $\boldsymbol{p}_t^m$, the two features are fused to form an intermediate video representation $\boldsymbol{u}_t$. Since words in a question may correlate with appearance and motion in different intensities, the model uses the current word to assign scores to both channels of features and fuses them as follows:

$$\begin{aligned} s_t^a, s_t^m &= softmax(\boldsymbol{W}_m \boldsymbol{x}_t + \boldsymbol{b}_m) \\ \boldsymbol{u}_t &= s_t^a \boldsymbol{p}_t^a + s_t^m \boldsymbol{p}_t^m \end{aligned} \tag{3}$$

The computed intensities are two scalars s_t^a, s_t^m, and the fused representation $\boldsymbol{u}_t$ absorbs information from both appearance and motion channels of the video based on current word.

Memory. The model processes one word of the question and performs two attention operations ATT_1 and ATT_2 at each timestep. We use LSTM_a to control the input of second attention operation and remember the attention history. The model has already generated the intermediate representation $\boldsymbol{u}_t$ of the video based on the current word. We now bring the question information into consideration. Since the hidden state $\boldsymbol{h}_t^q$ of the LSTM_q contains part of the question which has been processed already, the model uses it as well as the intermediate video representation $\boldsymbol{u}_t$ and the refined video representation $\boldsymbol{v}_{t-1}$ from last timestep to form the input of LSTM_a. The output $\boldsymbol{h}_t^a$ is used to perform second attention operation ATT_2.

Refine. After ATT_2 has been executed, the model generates the second weight $\boldsymbol{a}_t^2$ over $\boldsymbol{F}$. Both $\boldsymbol{a}_t^1$ and $\boldsymbol{a}_t^2$ are used to refine the attention. The detail of REF is represented in Figure 3 and can be formulated as follows:

$$\begin{aligned} \boldsymbol{a}_t &= (\boldsymbol{a}_t^1 + \boldsymbol{a}_t^2)/2 \\ \boldsymbol{g}_t &= \sum_{i=1}^{N} a_t^i \tanh(\boldsymbol{W}_f \boldsymbol{f}_i + \boldsymbol{b}_f) \\ \boldsymbol{v}_t &= CF(\boldsymbol{h}_t^q, \boldsymbol{g}_t) \end{aligned} \tag{4}$$

Here $\boldsymbol{g}_t$ actually includes $\boldsymbol{g}_t^a$ and $\boldsymbol{g}_t^m$ from appearance and motion, and $\boldsymbol{v}_t$ is the final fused representation of the video at timestep t.

With the above attention process, the model uses both fine-grained word information and coarse-grained question information to gradually refine the attention over appearance and motion features of the video. The attention with current word embedding can enhance the key word information which may be buried in single vector feature of the question, and the question information can give a more general guidance when fusing these features and refining the attention. After AMU processed all words of the question, the refined representation of video is generated which is most relevant and significant to answer the question.

3.3 Answer Generation

In this section, we present the two types of answer generation methods which are commonly used in the literature.

At timestep T, after the final word of the question being processed, we get the fused representation of video $\boldsymbol{v}_T$. We also have two more context information. The memory vector $\boldsymbol{c}_T^q$ of the question LSTM_q contains information about the question and the memory vector $\boldsymbol{c}_T^a$ of AMU contains information about attention history. We use these three aspects of information to generate the answer.

We can prepare a predefined answer set and the Generator can be a simple softmax classifier. The answer is chosen as follows:

$$answer = \arg\max \ softmax(\boldsymbol{W}_g(\boldsymbol{W}_x \boldsymbol{c}_T^q \cdot \boldsymbol{c}_T^a \cdot \boldsymbol{v}_T)) \tag{5}$$

The Generator can also be the LSTM network which is used commonly in sentence generation task. The question information $\boldsymbol{c}_T^q$ and attention history $\boldsymbol{c}_T^a$ can be used to initialize the LSTM network while the refined video representation $\boldsymbol{v}_T$ serves as its first

input. Each word of the answer can be generated as in Equation (5) except the choice is over the whole vocabulary.

Our proposed model is an end-to-end model which refines its attention over appearance and motion features of the video based on analyzing the question carefully. We adopt several experiments to evaluate our proposed model and the following section describes the concrete experiment settings.

4 EXPERIMENTS

In this section, we adopt several experiments on video question answering (VideoQA) datasets constructed from publicly available video captioning datasets. The results show the effectiveness of the proposed model.

4.1 Data Preparation

Since VideoQA is a relatively new task, there is no available public dataset. [9] presents a method which can generate question answer pairs from descriptions automatically. We generate two datasets based on this method by converting video captions in existing datasets to question answering pairs.

MSVD-QA. The dataset is based on Microsoft Research Video Description Corpus [3] which is used in many video captioning experiments. The MSVD-QA dataset has a total number of 1,970 video clips and 50,505 question answer pairs. We split the dataset based on videos that training set takes 61%, validation set takes 13%, and test set takes 26% of the total number of videos. Table 1 show the statistics of the MSVD-QA dataset.

MSRVTT-QA. The dataset is based on the MSR-VTT dataset [34] which is larger and has more complex scenes. The dataset contains 10K video clips and 243k question answer pairs. We follow the data split in MSR-VTT dataset which is 65% for training set, 5% for validation set and 30% for test set. Table 2 show the statistics of the MSRVTT-QA dataset.

4.2 Model details

We already present the framework of our proposed model in Section 3, here we show the concrete settings of the model adopted in experiments.

We use VGG network to extract features from the appearance channel, and C3D network to extract features from the motion channel. Given a video, we first sample 20 evenly distributed frames and clips respectively. Then VGG and C3D networks are applied to these frames and clips, from which the activations of last fully connected layer are extracted as the corresponding features. The number of features in both channels is 20 and the dimension of features is 4,096.

The question is transformed by the embedding layer, which is actually a matrix containing vector representations of all words in the vocabulary. We use the pre-trained 300-dimensional GloVe embedding [18] to initialize our Embedding layer. The GloVe used is trained based on *Wikipedia 2014* and *Gigaword 5*, which contains 400K vocabulary. We prune the GloVe embedding to match the size of vocabulary. For words in our vocabulary not appeared in GloVe, we average all of other existing word embeddings as their embeddings. We choose the size of LSTM_q to be 300 which matches the dimension of the word embedding.

Table 1: Statistic of the MSVD-QA.

	Video	QA pair	Question Type				
			what	who	how	when	where
Train	1,200	30,933	19,485	10,479	736	161	72
Val	250	6,415	3,995	2,168	185	51	16
Test	520	13,157	8,149	4,552	370	58	28
All	1,970	50,505	31,629	17,199	1,291	270	116

Table 2: Statistic of the MSRVTT-QA.

	Video	QA pair	Question Type				
			what	who	how	when	where
Train	6,513	158,581	108,792	43,592	4,067	1,626	504
Val	497	12,278	8,337	3,439	344	106	52
Test	2,990	72,821	49,869	20,385	1,640	677	250
All	10,000	243,680	166,998	67,416	6,051	2,409	806

The AMU performs attention operations based on representations of the video and the question. Since the two types of representation may vary in size, we choose 256 as the common dimension size for AMU. Both video features and word embeddings are mapped to this underlying common space before further computing. The size of the LSTM_a is also set to 256.

Although the open-ended answer is more natural, it needs more time and computation resources to adopt the experiment. Thus we use 1000-way softmax selection from the predefined answer set to generate the answer as many previous works [2, 43] do in question answering.

4.3 Baseline methods

For comparison, we choose to extend three models basically like [43] with some minor changes as our baselines. These models reflect the continuously strengthened power.

Extended VQA model (E-VQA). The model uses one LSTM network to encode all words in the question and another different LSTM network to encode the frames in the video. The representation of the question and video are then fused as a uniform representation, which is used to decode the answer. The model considers the sequenced nature of the video and question.

Extended Soft-Attention model (E-SA). The model first encodes the words in the question using a LSTM network, then the encoded representation is used to attend on features of video frames. Both the question and weighted video representation are used to generate the answer. The model adds the ability to select important frames based on the question.

Extended End-to-End Memory Networks (E-MN). The model uses the bidirectional LSTM network to update the frame representations of the video. The updated representations are mapped into the memory and the question representation is used to perform multiple inference steps to generate the answer. The model not only has the first two model's abilities but also augments and improves the inference procedure.

The above models do not have a dedicated way to process the appearance and motion information of the video. The features of

different channels are mean pooled between the corresponding frames and clips.

4.4 Training details

We implement our models and baseline models using TensorFlow [1], a framework of dataflow computation graph which serves deep learning method very well.

For both datasets, we choose the top K=1,000 most frequent answers as the answer set, which follows the setting in [2]. We also select several most frequent words from the training set of the dataset as vocabulary. The vocabulary size of MSVD-QA is 4,000 while for MSRVTT-QA it is 8,000.

We use mini-batch stochastic gradient descent to optimize the models and the Adam [13] with its default learning rate 0.001 as the optimizer. We use the batch size 32 for MSVD-QA and 64 for MSRVTT-QA. All of the models are trained at most 30 epochs with early stopping. To handle the questions of different lengths efficiently, we divide questions into several buckets based on the question length. The number of buckets used in MSVD-QA is 4. In MSRVTT-QA, the number is 5 since MSRVTT-QA has more longer questions. In each bucket, the questions are padded to the length of the longest question in that bucket. The loss function of all models is defined as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\log(p_{i,j}) + \lambda_1 \sum_{k} w_k^2 \tag{6}$$

The first part is the logarithmic loss, where N is the batch size, M is the number of possible answers, y_{ij} is a binary indicator of whether or not answer j is the correct answer for example i, and p_{ij} is the probability of assigning answer j to example i by the model. The second part is the L2 regularization on least squares where w_k represents the model weight and λ_1 is the hyperparameter controls the importance of the regularization. The regularization term is used to prevent the model from overfitting.

For our proposed model, since it uses the two channels explicitly, we also add another item in Equation (6) to encourage the model to utilize the features from all channels of videos. The item can be defined as:

$$\lambda_2 \sum_{i=1}^{N} |s_i^a - s_i^m| \tag{7}$$

Where N is the batch size, s_i^a and s_i^m denotes the importance score assigned to each channel for example i finally. The item is added to the original loss function with coefficient λ_2 when training our proposed model specifically.

4.5 Results and Analysis

We evaluate three baseline models and our proposed model in MSVD-QA and MSRVTT-QA. The accuracies are presented in Table 3 and Table 4.

For both datasets, our proposed model achieves higher overall accuracies than other baseline models, which indicates the effectiveness of our model. There are five types of questions in both datasets, thus we also report the accuracy in each question types. Our model achieves higher accuracies in question type *what* and *who* on both datasets, but the performance in other question types is slightly different. As shown in Table 1 and Table 2, questions of type *what* and *who* make up most of the questions in both datasets and hold the most diverse answers. The other three types of questions have very limited answers which are not general enough to reflect the performance of our model.

Table 3: Experiment results with MSVD-QA dataset.

Methods	what	who	how	when	where	ALL
E-VQA	0.097	0.422	**0.838**	**0.724**	**0.536**	0.233
E-SA	0.150	0.451	**0.838**	0.655	0.322	0.276
E-MN	0.129	0.465	0.803	0.707	0.500	0.267
Our Model	**0.206**	**0.475**	0.835	**0.724**	**0.536**	**0.320**

Table 4: Experiment results with MSRVTT-QA dataset.

Methods	what	who	how	when	where	ALL
E-VQA	0.189	0.387	0.835	0.705	0.292	0.264
E-SA	0.220	0.416	0.796	**0.731**	**0.332**	0.293
E-MN	0.234	0.418	**0.837**	0.708	0.276	0.304
Our Model	**0.262**	**0.430**	0.802	0.725	0.300	**0.325**

For qualitative analysis, we present several examples from both datasets. In Figure 4, the first two rows show four examples correctly answered by our model from MSVD-QA, and the last two rows show four examples from MSRVTT-QA. The video is represented in 5 concatenate frames that are sampled evenly distributed in the original video for saving space. We can see the model succeeds in answering questions that involve objects and actions. As have mentioned earlier in Section 4, the videos in MSRVTT-QA have more scene changes compared to videos in MSVD-QA, which can also be verified visually in the examples.

To have an intuitive understanding of how the proposed model answers the question, we also analyze the attention of two examples from MSRVTT-QA in Figure 5. The video is presented as 20 frames and 20 clips in each channel which exactly matches the settings in the experiment. We dive into the interior of the model and visualize the scores assigned to each channel on the left of the video. We also present the attention weights inside each channel generated by the model. In the first example, the question is *"what is squidward picking up?"* and the video is a cartoon. Our model gives the correct answer *"telephone"* for this question by focusing on the appearance channel and generating high attention weights on frames containing a telephone. In the second example, the question is *"what is a man doing?"* and the video is composed of three main scenes which are a moving car, a dash board and a man sitting in the car. We can see that since the question is asking about action, our model assigns a higher score to the motion channel than the appearance channel. In the motion channel, our model focuses on the first three clips which contain the motion of a moving car. Even the appearance channel is less used, our model still focuses on frames containing the man which is the subject of the question.

From the detailed analysis of refined attentions in both examples, we find that our proposed model exploits both channels of videos and selects the useful channel and features when answering the question.

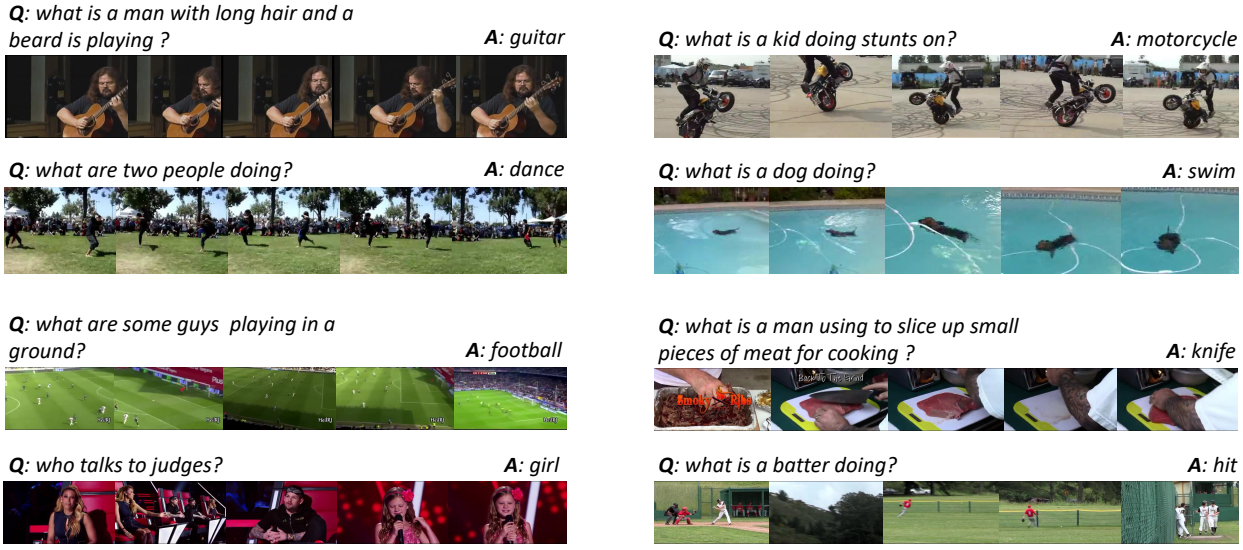

Figure 4: The correctly answered examples from both datasets.

Figure 5: Visualization of the attention for two examples. *a* stands for appearance and *m* stands for motion.

5 CONCLUSIONS

In this paper, we develop an end-to-end model which exploits the information from both appearance and motion channels in the video when performing the VideoQA task. To answer the question, the model extracts appearance and motion features of the video and gradually refines its attention over these features based on both coarse-grained question feature and fine-grained word feature. Our model can be easily extended to incorporate more information channels such as the text and audio channels. To evaluate our model, we perform extensive experiments on two datasets. The results show that our model can achieve better performance compared to other baseline models. We also perform several detailed analyses, showing that our model effectively focuses on the necessary information from different channels. In future, we will combine our video understanding method with user feedback (e.g., ratings and clicks) to user-oriented downstream applications, such as personalized recommendation [8, 30].

ACKNOWLEDGMENTS

This work was supported by the 973 program (2015CB352302), Zhejiang Natural Science Foundation (LZ17F020001), National Natural Science Foundation of China (61572431, 61602405, U1611461, U1611461), China Knowledge Centre for Engineering Sciences and Key Program of Zhejiang Province (2015C01027), National Key Research and Development Program of China (SQ2017YFGX030023).

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *ICCV*.

[3] David L Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *ACL*.

[4] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item-and Component-Level Attention. In *SIGIR*.

[5] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, and Tat-Seng Chua. 2017. SCA-CNN: Spatial and Channel-wise Attention in Convolutional Networks for Image Captioning. In *CVPR*.

[6] Rasool Fakoor, Abdel-rahman Mohamed, Margaret Mitchell, Sing Bing Kang, and Pushmeet Kohli. 2016. Memory-augmented Attention Modelling for Videos. *arXiv preprint arXiv:1611.02261* (2016).

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*.

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*.

[9] Michael Heilman and Noah A Smith. 2009. *Question generation via overgenerating transformations and ranking*. Technical Report. CARNEGIE-MELLON UNIV PITTSBURGH PA LANGUAGE TECHNOLOGIES INST.

[10] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3D convolutional neural networks for human action recognition. *TPAMI* (2013).

[11] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*.

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

[13] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[14] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*.

[15] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng. 2011. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

[17] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. 2016. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*.

[18] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *EMNLP*.

[19] Matthew J Roach, JD Mason, and Mark Pawlewski. 2001. Video genre classification using dynamics. In *ICASSP*.

[20] Paul Scovanner, Saad Ali, and Mubarak Shah. 2007. A 3-dimensional sift descriptor and its application to action recognition. In *ACM MM*.

[21] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229* (2013).

[22] Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *NIPS*.

[23] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556* (2014).

[24] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

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

[26] Subhashini Venugopalan, Huijuan Xu, Jeff Donahue, Marcus Rohrbach, Raymond Mooney, and Kate Saenko. 2014. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729* (2014).

[27] Meng Wang, Weijie Fu, Shijie Hao, Hengchang Liu, and Xindong Wu. 2017. Learning on Big Graph: Label Inference and Regularization with Anchor Hierarchy. *TKDE* (2017).

[28] Meng Wang, Richang Hong, Guangda Li, Zheng-Jun Zha, Shuicheng Yan, and Tat-Seng Chua. 2012. Event Driven Web Video Summarization by Tag Localization and Key-Shot Identification. *TMM* (2012).

[29] Meng Wang, Xian-Sheng Hua, Richang Hong, Jinhui Tang, Guo-Jun Qi, and Yan Song. 2009. Unified video annotation via multigraph learning. *TCSVT* (2009).

[30] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item Silk Road: Recommending Items from Information Domains to Social Users. In *SIGIR*.

[31] Qi Wu, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2016. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *CVPR*.

[32] Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *ICML*.

[33] Huijuan Xu and Kate Saenko. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*.

[34] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *CVPR*.

[35] Li-Qun Xu and Yongmin Li. 2003. Video classification using spatial-temporal features and PCA. In *ICME*.

[36] Xun Yang, Meng Wang, Richang Hong, Qi Tian, and Yong Rui. 2017. Enhancing Person Re-identification in a Self-trained Subspace. *arXiv preprint arXiv:1704.06020* (2017).

[37] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *CVPR*.

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

[39] Yunan Ye, Zhou Zhao, Yimeng Li, Long Chen, Jun Xiao, and Yueting Zhuang. 2017. Video Question Answering via Attributed-Augmented Attention Network Learning. In *SIGIR*.

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

[41] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. 2015. Beyond short snippets: Deep networks for video classification. In *CVPR*.

[42] Mihai Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. 2016. Spatio-Temporal Attention Models for Grounded Video Captioning. In *ACCV*.

[43] Kuo-Hao Zeng, Tseng-Hung Chen, Ching-Yao Chuang, Yuan-Hong Liao, Juan Carlos Niebles, and Min Sun. 2017. Leveraging Video Descriptions to Learn Video Question Answering. In *AAAI*.

[44] Hanwang Zhang, Xindi Shang, Huanbo Luan, Meng Wang, and Tat-Seng Chua. 2016. Learning from collective intelligence: Feature learning using social images and tags. *TOMM* (2016).

[45] Hanwang Zhang, Zheng-Jun Zha, Yang Yang, Shuicheng Yan, Yue Gao, and Tat-Seng Chua. 2013. Attribute-augmented semantic hierarchy: towards bridging semantic gap and intention gap in image retrieval. In *ACM MM*.

[46] Songyang Zhang, Xiaoming Liu, and Jun Xiao. 2017. On geometric features for skeleton-based action recognition using multilayer LSTM networks. In *WACV*.

[47] Zhou Zhao, Lu Hanqing, Deng Cai, Xiaofei He, and Yueting Zhuang. 2016. Partial Multi-Modal Sparse Coding via Adaptive Similarity Structure Regularization. In *ACM MM*.

[48] Zhou Zhao, Xiaofei He, Deng Cai, Lijun Zhang, Wilfred Ng, and Yueting Zhuang. 2016. Graph Regularized Feature Selection with Data Reconstruction. In *TKDE*.

[49] Zhou Zhao, Jinghao Lin, Xinghua Jiang, Deng Cai, Xiaofei He, and Yueting Zhuang. 2017. Video Question Answering via Hierarchical Dual-Level Attention Network Learning. In *ACM MM*.

[50] Zhou Zhao, Qifan Yang, Deng Cai, Xiaofei He, and Yueting Zhuang. 2017. Video Question Answering via Hierarchical Spatio-Temporal Attention Networks. In *IJCAI*.

[51] Linchao Zhu, Zhongwen Xu, Yi Yang, and Alexander G Hauptmann. 2015. Uncovering temporal context for video question and answering. *arXiv preprint arXiv:1511.04670* (2015).