Int J Comput Vis (2017) 124:409–421
DOI 10.1007/s11263-017-1033-7

Uncovering the Temporal Context for Video Question Answering

Linchao Zhu[1] · **Zhongwen Xu**[1] · **Yi Yang**[1] · **Alexander G. Hauptmann**[2]

Received: 18 July 2016 / Accepted: 4 July 2017 / Published online: 13 July 2017

Abstract In this work, we introduce Video Question Answering in the temporal domain to infer the past, describe the present and predict the future. We present an encoder–decoder approach using Recurrent Neural Networks to learn the temporal structures of videos and introduce a dual-channel ranking loss to answer multiple-choice questions. We explore approaches for finer understanding of video content using the question form of "fill-in-the-blank", and collect our Video Context QA dataset consisting of 109,895 video clips with a total duration of more than 1000 h from existing TACoS, MPII-MD and MEDTest 14 datasets. In addition, 390,744 corresponding questions are generated from annotations. Extensive experiments demonstrate that our approach significantly outperforms the compared baselines.

Keywords Video sequence modeling · Video question answering · Video prediction · Cross-media

1 Introduction

Current research into image analysis is gradually extending beyond recognition (Krizhevsky et al. 2012) and detection (Girshick et al. 2014). There is increasing interest in achieving deeper understanding of cross-media content by jointly modeling images and natural language. Early works of cross-media analysis only used hand-crafted features (Yang et al. 2009). As Convolutional Neural Networks (ConvNets) have raised the bar in image classification and detection tasks (Girshick et al. 2014; Ioffe and Szegedy 2015; Szegedy et al. 2015), Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997), play a key role in visual description tasks, such as image captioning (Donahue et al. 2015; Vinyals et al. 2015; Xu et al. 2015a). Image Question Answering (Image QA), which is one step beyond image captioning and requires an extra layer of interaction between humans and computers, has recently started to attract research attention (Antol et al. 2015; Gao et al. 2015; Malinowski et al. 2015).

In the area of video analysis, a small number of recent systems have been proposed for video captioning (Rohrbach et al. 2013; Venugopalan et al. 2015; Yao et al. 2015; Pan et al. 2016). These methods have demonstrated promising performance in describing a video by a single short sentence. Similar to image captioning, video captioning may not be as intelligent as desired, especially when we are only concerned with a particular section or object in the video (Antol et al. 2015). In addition, it lacks interaction between the computer and the user (Gao et al. 2015).

A MovieQA dataset (Tapaswi et al. 2015) was recently released focusing on story comprehension based on both movie clips and texts. The dataset contains some questions regarding "Why" and "How" about something. It is a challenging task and can only be resolved by exploiting visual and textual information. However, the task of MovieQA is basically to answer questions about a movie clip without any video context information. In this paper, we focus on Video Question Answering (Video QA) in the temporal domain. Different from MovieQA (Tapaswi et al. 2015), our Video QA task focuses on video temporal context understanding and thus consists of three subtasks, which are describing the

Communicated by Bernt Schiele.

✉ Yi Yang
yee.i.yang@gmail.com

[1] CAI, University of Technology Sydney, Sydney, NSW, Australia

[2] SCS, Carnegie Mellon University, Pittsburgh, PA, USA

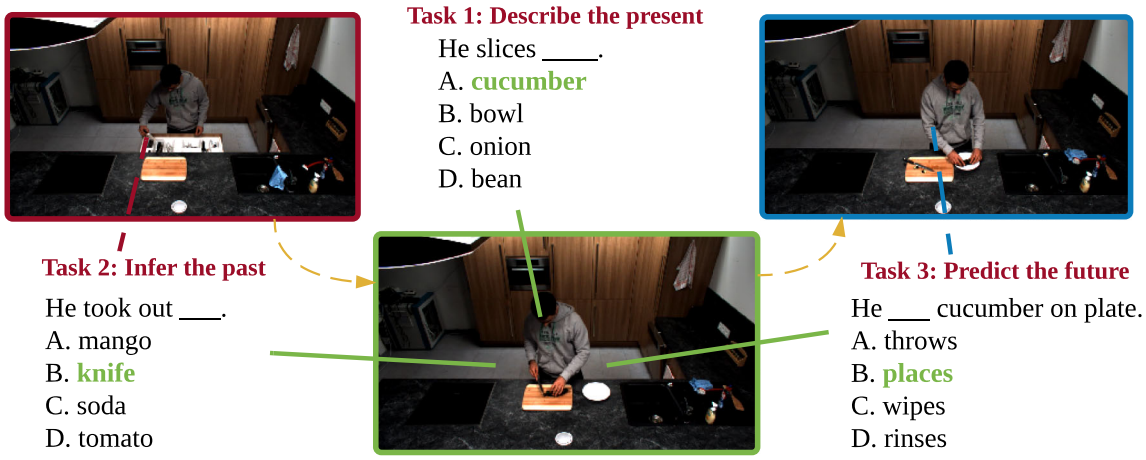

Fig. 1 Questions and answers about the past, the present and the future. Our system includes three subtasks, which infer the *past*, describe the *present*, and predict the *future*, while *only the current frames are observable*. Best viewed in *color* (Color figure online)

present, inferring the past and the future. As shown in Fig. 1, if we see a man slicing cucumber on a cutting board, we can infer that he *previously* took a knife from the drawer, and predict that he will put the cucumber slices on a plate *afterwards*. As with Image QA, Video QA requires a finer understanding of videos and sentences than video captioning where questions about local objects and actions are required to be answered. Despite the success of video captioning in (Venugopalan et al. 2015; Yao et al. 2015; Pan et al. 2016), a number of research challenges remain unsolved, which means these methods are not readily applicable to Video QA.

Firstly, a Video QA system should explore more knowledge beyond the visual information and coarse sentence annotations because it requires a finer understanding of video content and questions. For the sake of video captioning, existing systems train LSTM models based on the video content and associated coarse sentence annotations alone. Because the size of the description embedding matrix is very large and many words usually appear fewer than 10 times in all descriptions, the model overfits easily. A recent study (Lin and Parikh 2015) found that visual information and textual information are mutually beneficial. We developed a new way to approach Video QA, by appropriately integrating information, including sentences, words, and visual cues, within a joint learning framework to maximize the mutual benefit. Using this method, external knowledge bases [e.g. BookCorpus (Zhu et al. 2015) and Google News (Mikolov et al. 2013)] can be readily incorporated. Because the external knowledge bases reflect the underlying correlations between related entities, our approach is able to to better parse questions and answers.

Secondly, a Video QA system should be capable of reasoning across video frames, including inferring the past, describing the present, and predicting the future, all of which are strongly correlated. Gated Recurrent Unit (GRU) (Cho et al. 2015) has recently demonstrated promising performance on sequence modeling tasks, partially because it has a simpler neural structure than LSTM. On top of GRU, we propose an *encoder–decoder* approach with a *dual-channel ranking loss* to learn three video representations, one for each Video QA subtask, i.e., past inference, present description, and future prediction. One appealing feature of our approach is that the encoder–decoder approach is able to model a wider range of temporal information, and the reduced number of weight parameters in GRU makes it more robust to overfitting in temporal modeling. Further, the approach eliminates the need to create a large number of labels to train the sequence model by embedding visual features in a semantic space.

Thirdly, a well-defined quantitative evaluation metric and datasets from different domains to track progress of this important research (Malinowski et al. 2015; Malinowski and Fritz 2014; Antol et al. 2015) are required. Manually providing groundtruth for a large number of videos is extremely human labor intensive. BLEU (Papineni et al. 2002) has been widely used as an evaluation metric for image captioning, but a few research papers and competition reports have indicated that BLEU is not a reliable metric and cannot reflect human judgment (Kulkarni et al. 2011; Vedantam et al. 2015). Following Lin and Parikh (2015), Yu et al. (2015), we evaluated our question answering approach in the form of "fill-in-the-blank" (FITB) multiple choice responses. We utilized over 100,000 real-world videos clips from existing TACoS, MPII-MD and MEDTest 14 datasets, and generated 400,000 designed questions with more than 1,000,000 candidate answers. This dataset will be released to the public and can be used as the benchmark for this research. The main advantage is that it is more convenient for quantitative evaluation than free-style question answering.

In this paper, we propose a new framework for Video QA by carefully addressing the three aforementioned challenges. The rest of this paper is organized as follows. After introduc-

ing related works, we detail the large scale dataset we have collected for Video QA tasks. We then present our video temporal structure modeling approach and dual-channel learning-to-rank method for question answering. Extensive experiments are conducted to validate our approach. We will release the source code and models for reproducibility upon acceptance.

2 Related Works

Neural networks in video analysis Many video feature learning methods based on ConvNets have recently been proposed. Simonyan and Zisserman (2014) proposed the use of optical flow images extracted from videos as the inputs to train ConvNets. Along with the ordinal RGB stream, two-stream ConvNets can achieve comparable performance with the state-of-the-art hand-crafted feature improved Dense Trajectories (Wang et al. 2013). Tran et al. (2015) proposed 3D ConvNets which capture temporal dynamics in video clips without the very time-consuming optical flow extraction procedure. Xu et al. (2015b) adapted the ConvNet frame-level features by VLAD pooling over the timestamps to generate video representation, which has great advantages over traditional average pooling. A general sequence to sequence framework *encoder–decoder* was introduced by Sutskever et al. (2014) which utilizes a multilayered RNN to encode a sequence of inputs into one hidden state, following which another RNN takes the encoded state as inputs and decodes it into a sequence of outputs. Srivastava et al. (2015) extended this general model to learn features from consecutive frames and proposed a composite model for unsupervised LSTM autoencoder. Gan et al. (2016) proposed a method of video-text joint modeling for zero-shot action recognition.

Bridging vision and language: captioning and question answering There is increased interest in the field of multimodal learning for bridging computer vision and natural language understanding (Donahue et al. 2015; Karpathy and Fei-Fei 2015; Yu and Siskind 2013; Venugopalan et al. 2015; Vinyals et al. 2015; Yao et al. 2015; Ordonez et al. 2015; Yan et al. 2016). Captioning is a particular popular task, and LSTM is heavily used as a recurrent neural network language model to automatically generate a sequence of words conditioned on the visual features, inspired by the general recurrent encoder–decoder framework (Sutskever et al. 2014). However, the captioning task only generates a generic description for the entire image or video clip, and it is difficult to evaluate the quality of the sentences generated; that is, it is difficult to judge whether one description is better than another. In addition, designing a proper metric for visual captioning which can reflect human judgment (Elliott and Keller 2014; Vedantam et al. 2015) is still an open research problem. Mao et al. (2015) proposed a Speaker-Listener method to generate unambiguous descriptions. In this work, we instead focus on a more fine-grained description of video content, and our method is simple to evaluate in multiple-choice form, i.e., by selection of the correct answer. A number of QA datasets and systems have recently been developed on images (Antol et al. 2015; Gao et al. 2015; Malinowski and Fritz 2014). Gao et al. (2015) used a complex dataset with free-style multilingual question-answer pairs; however it is difficult to evaluate the answers, and human judgement is usually required. Lin and Parikh (2015) introduced an interesting multiple-choice fill-in-the-blank question answering task on abstract scenes, and Yu et al. (2015) applied the task to natural images using various question templates. Wu et al. (2016) incorporated an external knowledge base, i.e., DBpedia (Auer et al. 2007), to facilitate the image QA task by querying the relevant information from the knowledge base. In contrast, we leverage the external knowledge base by directly using the pre-trained models based on large datasets, e.g., BookCorpus (Zhu et al. 2015), rather than training the language model from scratch. Unlike still images, video analysis can utilize the temporal information across frames, along with object and scene information. The richer structural information in videos potentially enables better understanding of the visual content while at the same time imposing challenges.

Video Question Answering and temporal structure reasoning One of the recent works on video-based question answering is Tu et al. (2014), which built a query answering system based on a joint parsing graph from both text and videos. However, Tu et al. restricted their model to surveillance videos of predefined structure, which cannot deal with open-ended questions. MovieQA (Tapaswi et al. 2015) uses movies as a single source, which are produced by professionals in controlled environment. Differently, we aim to deal with videos which could be produced by anyone in the wild. Therefore, we collect videos from various sources, e.g., cooking scenarios, unconstrained web videos from YouTube, based on which a model is trained to capture the dynamic temporal structure of unconstrained videos. Action forecasting, from the aspect of temporal structure learning, was initially studied by Vondrick et al. (2015). To predict the potential actions, Vondrick et al. proposed to use a regression loss built upon a ConvNet and forecast limited categories of actions and objects in a very short period, e.g., 1 s. In contrast, we utilize a more flexible encoder–decoder framework, modeling a wider range of temporal information, and we mainly focus on multiple-choice question answering tasks in the temporal domain, which goes well beyond standard visual recognition.

3 Dataset Collection and Task Definitions

The goal of our work is to present a Video QA system in the temporal domain that can infer the past, describe the present

and predict the future. We first describe our Video Context QA dataset (VCQA) collection and the method of automatically generating template questions in Sect. 3.1. Task definitions and dataset analysis will be discussed in Sect. 3.2.

3.1 Dataset and QA Pair Generation

We utilized more than 100,000 videos and 400,000 questions in total, while QA pairs were generated from existing datasets in different domains: a cooking scenario, DVD movies, and web videos:

1. *TACoS* (Regneri et al. 2013) The TACoS dataset consists of 127 long videos with a total of 18,227 annotations in the *cooking scenario*. It provides multiple sentence descriptions at a fine-grained level, i.e., for each short clip in each long video.
2. *MPII-MD* (Rohrbach et al. 2015) MPII-MD is collected from *DVD movies* where descriptions are generated from movie scripts semi-automatically. The dataset contains 68,375 clips and one annotation on average is provided for each clip.
3. *TRECVID MEDTest 14* (MED 2014) TRECVID MEDTest 14 is a complex event wild video dataset collected from *web* hosting services such as YouTube. Videos in the dataset total some 1300 h in duration. The videos are untrimmed and an annotation is provided for each long video which can be regarded as a coarse high-level summarization compared to the TACoS and MPII-MD datasets.

Question template generation Ren et al. (2015) transform image descriptions to questions with only four question types, i.e., objects, number, color and location. It is difficult to automatically transform descriptions into free-from QA, which is still an open-ended topic (Ren et al. 2015). In contrast, we use the FITB form where the questions are transformed in an easier way and the question types are more diverse. We use the Stanford NLP Parser (Klein and Manning 2003) to obtain the syntactic structures of original video descriptions. We divide the questions into three categories, nouns (objects like food, animals, plants), verbs (actions) and phrases (short phrases, e.g., "play computer game"). Question templates are subsequently generated from noun phrases (NP) and verb phrases (VP). Multiple words can be dropped to form the questions. During template generation, we eliminate prepositional phrases as they are mostly subjective. We use WordNet[1] and NLTK[2] toolkits to identify word categories and choose a set of categories listed in Table 1. We

[1] https://wordnet.princeton.edu.

[2] http://www.nltk.org/.

Table 1 List of categories and number of collected words in three datasets

Datasets	TACoS	MPII-MD	MEDTest 14	Combined
Verbs	268	869	671	2925
Phrases	964	220	418	5927
Animals	–	63	98	352
Food/plant	62	129	174	598
Other objects	134	896	726	2093

The right column shows the total number of words and phrases collected, including those from image domains such as MS COCO (Lin et al. 2014)

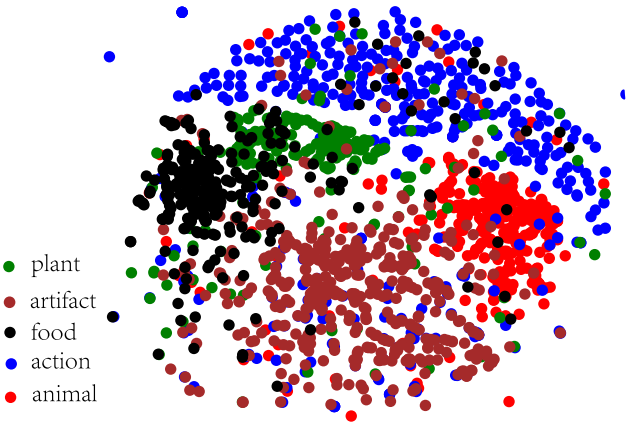

Fig. 2 t-SNE visualization of word embeddings for each category learned from word2vec model

visualize the distribution of words in each category using t-SNE (Van der Maaten and Hinton 2008) in Fig. 2. It shows that categories can be separated, where actions and objects have a clear margin.

Candidate answer generation After question template generation, we obtained a question template and a correct answer, where distractors are still required to form the multiple-choice FITB dataset. We designed two different levels of difficulty for answering questions by altering the similarities between the correct answer and the distractors.

For each question in the easy task, we randomly chose three distractors in the same category from the same dataset. We thus have four candidates for the easy tasks. Words like "person" or "man" were filtered in advance, and words with a frequency of less than 10 were filtered following common practice.

Video clips in the same dataset can have totally different scenes, e.g., the web videos are very diverse in MEDTest 14 dataset. To generate more difficult questions, we selected hard negative distractors from phrases that are similar to the correct answer. We collected distractors not only from the video datasets, but also used annotations from Flickr8K (Hodosh et al. 2013), Flickr30K (Young et al. 2014) and MS COCO (Lin et al. 2014) as description sources for the similarity comparison. We first parsed the annotations using

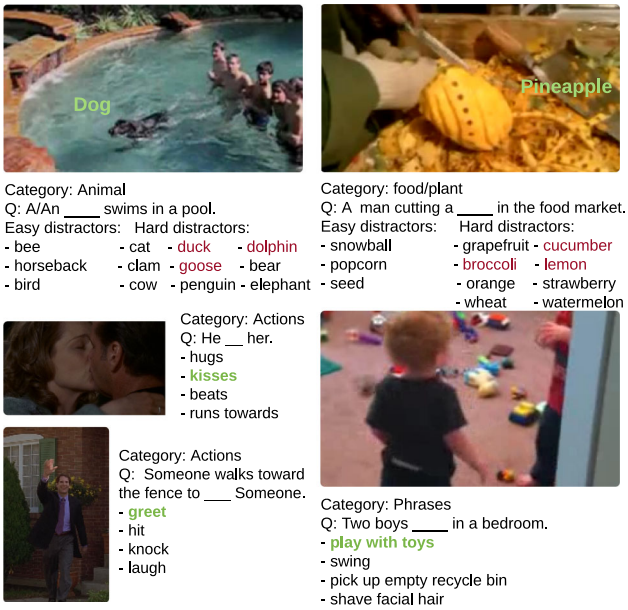

Fig. 3 Examples of QA pairs for different categories and levels of difficulty. The words *colored* in *green* are correct answers, and the difficult candidates are *marked in red* (Color figure online)

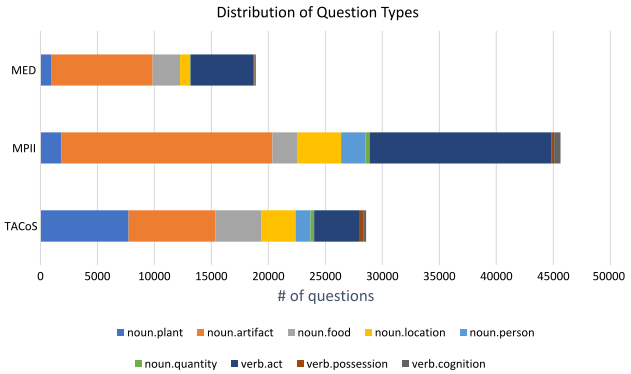

Fig. 4 Distribution of question types for each dataset

the method described above and gathered about 8000 phrases in total, resulting in an average length of 6.6 words per phrase. After the preprocessing, we encoded both the correct answers and the distractors with word2vec (Mikolov et al. 2013) and measured the similarity with cosine distance. For candidate phrases with length more than one, we average the word2vec representation of each word (Lebret et al. 2015; Lin and Parikh 2015). To avoid the ambiguity between the distractors and the correct answer (distractors should not be too close to the correct answer), we manually set an upper bound for the similarity between them. We chose the best threshold by sampling a few thousand candidates and checking the ambiguity by human labor. We discarded phrases that are too similar to the correct answer, and selected nine most similar distractors from the remaining phrases. We thus have ten candidates for the difficult task. We show examples of QA pairs of different categories and level of difficulty in Fig. 3.

3.2 Task Definitions and Analysis

In addition to describing the current clip, we introduce another two tasks which respectively infer the past and anticipate the future. In the task of describing the present, we use all three datasets for evaluation. For the other two tasks of past inference and future prediction, we perform experiments on the TACoS and MPII-MD datasets only because they are annotated in fine-grained clips. In these tasks, questions about the previous or next clip need to be answered for the given video clip. Note that for tasks of describing the past or the future, only the current clip is given and the model has to reason temporal structures based on the given clip. We restrict the past and future so that they are not too far away from the current clip and typically we choose the clip immediately before or after the given clip, where the time interval is less than 10 s.

We introduce two levels of questions for each task. For simplicity, we denote our tasks as *Past-Easy*, *Present-Easy*, *Future-Easy*, *Past-Hard*, *Present-Hard* and *Future-Hard*. We randomly partitioned the dataset into three non-overlapping subsets, one for training the models, one for validation (hyper-parameter tuning), and one for testing the performance. As the performance may vary in accordance with different subset partitions, we randomly partitioned the dataset three times independently. We thus have three splits for each task. The models are trained individually on the training set of each split. The parameters are not shared between splits.

We now show the statistics of our dataset.

Question types We visualize the distribution of question types in Fig. 4. It shows that there are more questions about objects than actions. The distribution of each question type also varies across different datasets, e.g., there are more questions about food and plant in the TACoS dataset.

Popular questions and answer We show the top-5 popular questions and answers in Table 2. Each dataset has different popular questions and answers, from which we observe that the TACoS dataset asks more about cooking and it is less diverse than the other two. Questions such as "someone ___.", "people ___." can have richer answers than "the person rinse the ___.".

Question and answer lengths The lengths of questions and answers are shown in Figs. 5 and 6, respectively. We can see that the length of most questions are in the range from 5 to 12, while most answers only have one word and in TACoS and MPII-MD, about 10% answers have two words.

Table 2 Top-5 most popular questions and answers in each dataset

Rank	TACoS (%)	MPII-MD (%)	MEDTest 14 (%)
Questions			
1	The person get out a ___. (1.11)	Someone ___. (1.08)	Child ___ football. (0.14)
2	The person ___ a knife. (0.52)	He ___. (0.26)	Guy ___ ping pong. (0.07)
3	The person rinse the ___. (0.46)	Someone ___ someone. (0.17)	A man shave his ___. (0.06)
4	He ___ cut board. (0.32)	Someone ___ up. (0.10)	One ___ lady dancing indoors. (0.04)
5	The person ___ a plate. (0.24)	Someone ___ his head. (0.05)	People ___. (0.04)
Answers			
1	Take (0.05)	Look (0.02)	Play (0.04)
2	Cut (0.04)	Eye (0.01)	Dog (0.03)
3	Wash (0.03)	Hand (0.01)	Dance (0.03)
4	Rinse (0.02)	Head (0.01)	Kid (0.02)
5	Orange (0.02)	Sit (0.01)	Baby (0.01)

The numbers in the parentheses are the percentages of questions and answers

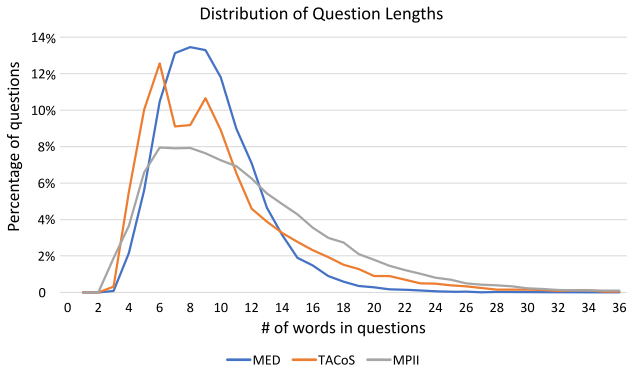

Fig. 5 Distribution of question lengths for each dataset

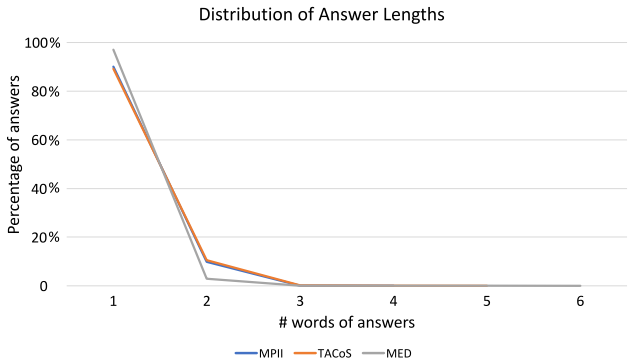

Fig. 6 Distribution of answer lengths for each dataset

4 The Proposed Approach

To answer questions about present, past and future, we first introduce an encoder–decoder framework to represent context. We then map the visual representation to a semantic embedding space and learn to rank the candidate answers.

4.1 Learning to Represent Video Sequences

In this section, we describe our model for learning temporal context. We present an encoder–decoder framework using Gated Recurrent Unit (GRU) (Cho et al. 2015). Compared with LSTM (Hochreiter and Schmidhuber 1997), GRU is conceptually simpler with only two gates (update gate and reset gate) and no memory cells, while the performance on the sequence modeling task is as good as LSTM (Chung et al. 2014). Note that we trained our model with LSTM as well, but it performs worse than the model trained with GRU. With GRU, we can achieve mAP of 24.9% on the MEDTest 14 100Ex classification task, whereas we can only achieve 20.4% with LSTM.

Gated Recurrent Unit Denote $f_i^1, f_i^2, \ldots, f_i^N$ as the frames in a video v_i, where N is the number of frames sampled from the video. At each step t, the encoder generates a hidden state $\mathbf{h}_i^t$, which can be regarded as the representation of sequence $f_i^1, f_i^2, \ldots, f_i^t$. Thus the state of $\mathbf{h}_i^N$ encodes the whole sequence of frames. States in GRU (Cho et al. 2015) are calculated as follows (the video subscript i is dropped for simplicity):

$$\mathbf{r}^t = \sigma(W_{xr}\mathbf{x}^t + W_{hr}\mathbf{h}^{t-1}), \quad (1)$$

$$\mathbf{z}^t = \sigma(W_{xz}\mathbf{x}^t + W_{hz}\mathbf{h}^{t-1}), \quad (2)$$

$$\bar{\mathbf{h}}^t = \tanh(W_{x\bar{h}}\mathbf{x}^t + W_{h\bar{h}}(\mathbf{r}^t \odot \mathbf{h}^{t-1})), \quad (3)$$

$$\mathbf{h}^t = (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t, \quad (4)$$

where $\mathbf{x}^t$ is the input, $\mathbf{r}^t$ is the reset gate, $\mathbf{z}^t$ is the update gate, $\mathbf{h}^t$ is the proposed state and $\odot$ is element-wise multiplication. We use the same architecture for the decoder as for the encoder, but its hidden state of $\mathbf{h}^0$ is initialized with the hidden state of the last time step N in the encoder. We

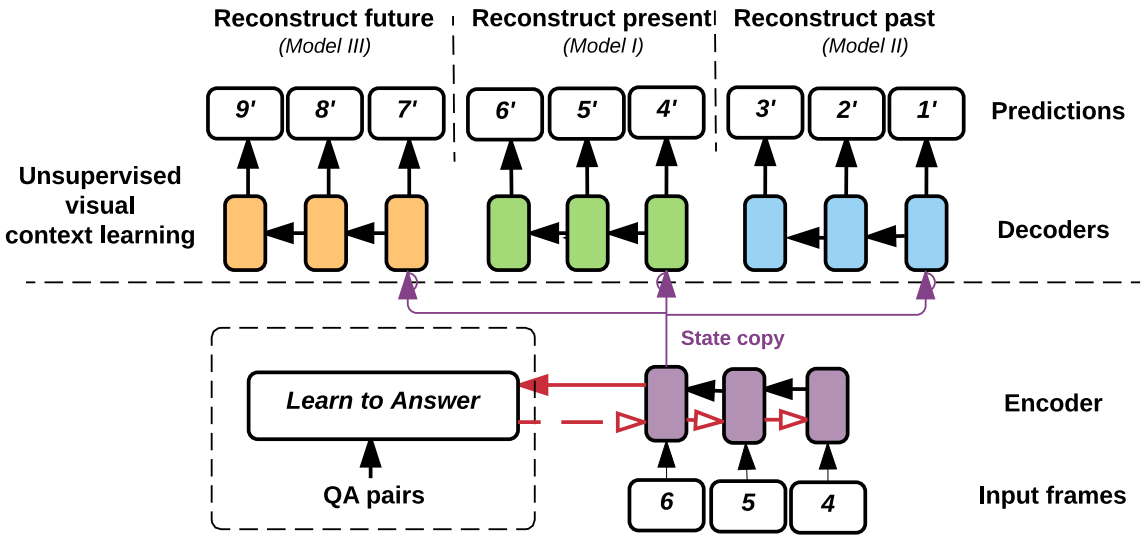

Fig. 7 The encoder–decoder model (*top*): encoder state of last time step is passed to three decoders for reconstruction. Learn to answer questions (*bottom*): encoder state of last time step is passed to the ranking module which selects an answer based on the visual information

construct our GRU encoder–decoder model (Fig. 7) in a similar fashion to Srivastava et al. (2015). Besides reconstructing the input frames, we also train another two models which are asked to reconstruct the future frames (Fig. 7 left) and past frames (Fig. 7 right), respectively. Our proposed models are capable of learning good features as the network is optimized by minimizing the reconstruction error. To achieve good reconstruction, representation passed to the decoder should retain high level abstraction of the target sequence. Note that our three models are learned separately, and the encoder and decoder weights are not shared across models of past, present and future.

Training We first train the encoder–decoder models in an unsupervised way using videos collected from a subset of the MED dataset (MED 2014) (excluding the MEDTest 13 and MEDTest 14 videos) which consists of 35,805 videos having a duration of 1300 h. The reason for choosing MED dataset as a source for temporal context learning is that videos in the MED dataset are longer in duration and contain complex and profound events, actions and objects for learning. We collect additional data to our target task datasets for the purpose of learning a more powerful model, and practically, it is difficult to train a model from scratch using such a small dataset as TACoS, which has only 127 cooking videos. As the video frames have high correlations in short range, we sample frames at the frame rate of 1 fps. We use a time span of 30 s and set the unroll length T to 30 for the present model (Model 1), and 15 for both the past model (Model 2) and the future model (Model 3).

For the input to the GRU model, we use ConvNet features extracted from GoogLeNet (Szegedy et al. 2015) with Batch Normalization (Ioffe and Szegedy 2015), which was trained from scratch with the ImageNet 2012 dataset (Russakovsky et al. 2015) and we keep the ConvNets part frozen during RNN training.

We now explain our network structures and training process in detail. As three models are trained with the same hyper-parameters, we take Model 1 as an example. In our case, reconstruction error is measured by ℓ_2 distance between the predicted representation and the target sequence. We reverse the target sequences in the present reconstruction scenario which, as indicated in Sutskever et al. (2014), reduces the path of the gradient flow. We set the size of the GRU units to 1024 and two GRU layers are stacked. Our decoders are conditioned on the inputs, and we apply Dropout with the rate of 0.5 at connections between the first GRU layer and the second GRU layer as suggested by Zaremba et al. (2014) to improve the generalization of the neural network. We initialize $\mathbf{h}^0$ for the encoder with zeros, while the weights in the input transformation layer are initialized with a uniform distribution in $[-0.01, 0.01]$ and recurrent weights have uniform distribution in $[-0.05, 0.05]$. We set the mini-batch size to 64 and clip gradient element-wise at the norm of $1e{-}4$. Frame sequences from different videos are sampled in each mini-batch. The network is optimized by RMSprop (Tieleman and Hinton 2012), which scales the gradient by a running average of gradient norm. The model is trained by the Torch library (Collobert et al. 2011) on a single NVIDIA Tesla K20 GPU, and it takes approximately one day for the models to converge and complete the training.

Inference At inference time, we feed the ConvNet features extracted from GoogLeNet to the encoder, and obtain the video features from the hidden states. For each video clip, we initialize $\mathbf{h}^0$ to zeros, and pass the current hidden state to the next step until the last input. We then average the hidden states at each time step as the final representation.

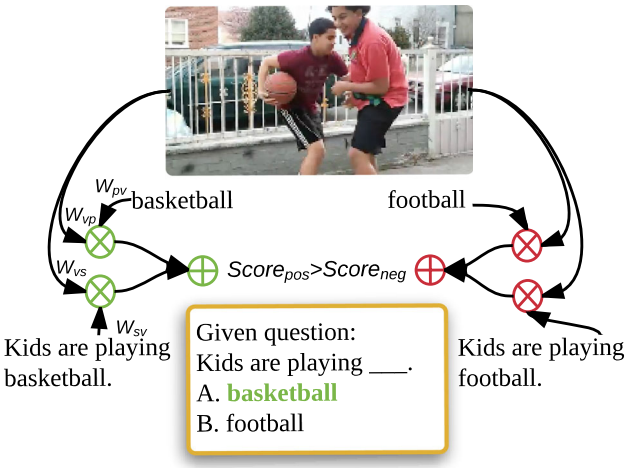

Fig. 8 Illustration of dual-channel learning to rank

4.2 Dual-Channel Learning to Rank

Visual information and textual information are mutually beneficial. We present the proposed dual-channel learning to rank algorithm by appropriately integrating information, including sentences, words, and visual cues, within a joint learning framework to maximize the mutual benefit. We jointly model two channels, i.e., word channel and sentence channel, for learning.

Kiros et al. (2015) recently proposed skip-thought vectors to encode a sentence into a compact vector. The model uses an RNN encoder to encode a sentence and another two RNN decoders are asked to reconstruct the previous sentence and the next sentence. It was trained using BookCorpus dataset (Zhu et al. 2015) which consists of 11,038 books, 74,004,228 sentences and 984,846,357 words. The skip-thought vectors model performs well on many different natural language processing (NLP) tasks. We utilize the combine-skip model to encode sentences. For more details, please refer to Kiros et al. (2015).

We first formulate the problem of multiple-choice question answering. Given N questions with blanks together with corresponding videos, and K candidate answers for each question, we denote each question as $q_i, i \in 1, \ldots, N$, candidate answers for question q_i as $p_{ij}, j \in 1, \ldots, K$ and the ground truth for question q_i as p'_i with index j'_i. For each question q_i, let s_{ij} be the sentence formed by filling the blank of question q_i with candidate p_{ij}. For example, filling in the template of "A/An ___ swims in a pool" shown in Fig. 3 with candidate "dog", we can form the sentence "A dog swims in a pool", and the false description "A horseback swims in a pool" is generated by "horseback".

Given q_i, we introduce a dual-channel ranking loss (also illustrated in Fig. 8) that is trained to produce higher similarity for the visual context and representation vector of the correct answer p'_i than other distractors $p_{ij}, j \neq j'_i$. We define our loss as:

$$\min_{\boldsymbol{\theta}} \sum_{\mathbf{v}} \sum_{j \in K, j \neq j'} \lambda \ell_{word} + (1-\lambda)\ell_{sent}, \lambda \in [0,1], \tag{5}$$

with

$$\begin{aligned} \ell_{word} &= \max(0, \alpha - \mathbf{v_p}^T \mathbf{p}_{j'} + \mathbf{v_p}^T \mathbf{p}_j), \\ \ell_{sent} &= \max(0, \beta - \mathbf{v_s}^T \mathbf{s}_{j'} + \mathbf{v_s}^T \mathbf{s}_j), \end{aligned} \tag{6}$$

where $\mathbf{v_p} = W_{vp}\mathbf{v}, \mathbf{v_s} = W_{vs}\mathbf{v}$ and $\mathbf{p}_j = W_{pv}\mathbf{y}_j, \mathbf{s}_j = W_{sv}\mathbf{z}_j$ (for simplicity we drop the subscript i). $\mathbf{v}$ is the vector learning from our GRU encoder–decoder model for video clip v_i, $\mathbf{y}_j$ is the average of word2vec vectors for each word in candidate p_{ij}, $\mathbf{z}_j$ is the skip-thought vector for description $\mathbf{s}_{ij}$. We constrain these feature representations to be in unit norm. $\boldsymbol{\theta}$ denotes all the transformation parameters that need to be learned in the model, W_{vs} and W_{vp} are transformations that map the visual representation to the semantic joint space, while W_{sv} and W_{pv} transform the semantic representation. Note that W_{xx} can be a linear transformation or multi-layer neural network with hidden units.

Training During the training procedure, we sample false terms from negative candidates and practically stop summing after finding the first margin-violating term (Frome et al. 2013). Empirically, we select a sentence embedding dimension of 500 and word embedding of 300. The model is trained by stochastic gradient descent (SGD) by simply setting the learning rate η as 0.01 and the momentum as 0.9. In practice, we set the margins α and β to 0.2, and λ is cross-validated on the held-out validation set.

Inference We learn the weight of the transformations at the training stage and at inference time, we calculate the following score for each candidate,

$$score = \lambda \mathbf{v_p}^T \mathbf{p}_j + (1-\lambda)\mathbf{v_s}^T \mathbf{s}_j \tag{7}$$

The candidate with the highest score is our answer.

5 Experiments

5.1 Evaluation of Describing the Present

In this section, we evaluate our model in the task of describing the present. We first present text-only baselines to show that visual information is an important cue in this task. We also demonstrate the effectiveness of our ranking method by comparing it with Canonical Correlation Analysis (CCA), normalized CCA (nCCA) and then conduct evaluations

Table 3 Comparison between our model and other baselines in the *Present-Easy* task

Methods	TACoS (%)	MPII-MD (%)	MEDTest 14 (%)
Simple skip-thought	35.4	25.8	26.4
Simple word2vec	36.9	34.3	35.1
LSTM Generator	54.6	60.3	69.3
GRU Generator	54.4	61.0	68.9
Question type Prior	63.3	47.7	63.0
CCA	65.1	41.6	63.2
nCCA	69.2	42.8	64.2
Post-hoc	67.0	35.1	55.7
Visual + Answer-only	63.1	53.8	73.0
Our dual-channel loss	76.3	71.9	81.0

Except the text-only baselines, "Mean-GoogLeNet" is used as visual features for all approaches

of dual-channel learning. We then show the biases in the answers.

Text-only baselines We show that visual information is an important cue by presenting text-only baselines where only question templates and answers are provided. In the first baseline, we choose the candidate which is most similar to the question as our answer. We average word2vec representation for each word to get the phrase representation for both questions and candidates. We measure the similarity by using the dot product. The candidate with the highest score will be the answer. We call it the *Simple word2vec* baseline. We also encode the question and candidates with skip-thought vectors which is our *Simple skip-thought* baseline.

In the second baseline, we use a LSTM decoder to generate the probability of each word in the vocabulary while the encoder encodes words in the question (Kiros et al. 2015; Ren et al. 2015; Malinowski et al. 2015; Gao et al. 2015). We choose the answer with the highest probability among the candidates. We call it the *LSTM Generator* baseline. A *GRU Generator* can also be introduced by replacing the LSTM cell with GRU cell. We use RNN cell size of 512 and set the number of RNN layer to 1 in the experiment. We early stop the training process when the performance on the validation set does not improve.

We compare the baselines with our model in the *Present-Easy* task and the results are shown in Table 3. It shows our visual model outperforms the text-only baselines with a large margin. The *Simple* baseline performs much worse than other methods as our dataset is constructed by choosing similar candidates. The results show that visual information is important in the task. We also observe that *Simple word2vec* performs better than *Simple skip-thought* which indicates that skip-thought might not encode short phrases effectively. *LSTM Generator* and *GRU Generator* have comparable results when modeling language.

Comparison with models with visual cue We first compare our dual-channel ranking approach with CCA which computes the direction of maximal correlation between a pair of multi-dimensional variables. Gong et al. (2014) normalized the correlation by dimension reduction before linear CCA and Yu et al. (2015) found it also beneficial to image QA. To learn CCA, we separately train two embedding layers. The first CCA maps the sentence description to the visual semantic joint-embedding space and the second CCA maps the correct answer to the joint space. To answer multiple-choice questions, we embed each candidate and select the answer that is most similar to the video clip using Eq. 7. We conduct cross-validation for choosing the weight to combine two embeddings. For nCCA, we also cross validated the dimension of the projected space on the validation set.

We restrict the input features to be the same for both methods. For visual representation, we average the frame-level features extracted from the last fully connected layer of GoogLeNet. For semantic representation, we use the same method as described in Sect. 4.2, where sentences are encoded by skip-thought vectors, and word2vec is used for word representation.

Note that in CCA, the two embedding matrices are learned separately at training time while the weights of two embeddings are introduced at the validation stage. The method of late fusing sentence and word descriptions is different from our dual-channel ranking approach, which integrates sentence and word representations during training time and learns to adjust the embeddings accordingly. We demonstrate the effectiveness of our dual-channel ranking method in Table 3.

As can be seen, nCCA consistently better than CCA and our method outperforms nCCA by a large margin. We believe it is because our objective function learns to integrate two representations, while nCCA uses a fixed embedding matrix during semantic weight learning. In addition, nCCA eliminates negative terms during training, and as multiple-choice question answering is required to select an answer from candidates at testing time, ranking loss is more suitable for modeling the problem.

Following Yu et al. (2015), we also use the video captioning model to generate the description of the video. We then compute the cosine similarity between the skip-thought representation of the description generated and the description filled by all the candidates. The most similar candidate is the our answer. We call it the *Post-hoc* baseline. We use GRU cell of cell size 512 in the captioning model and the results are shown in Table 3. We observe that the *Post-hoc* baseline is comparable to CCA on TACoS dataset but worse on MPII-MD and MEDTest 14 datasets.

Effectiveness of dual-channel loss We now show the effectiveness of using two channels for learning. The result of how the integration of two representations influences perfor-

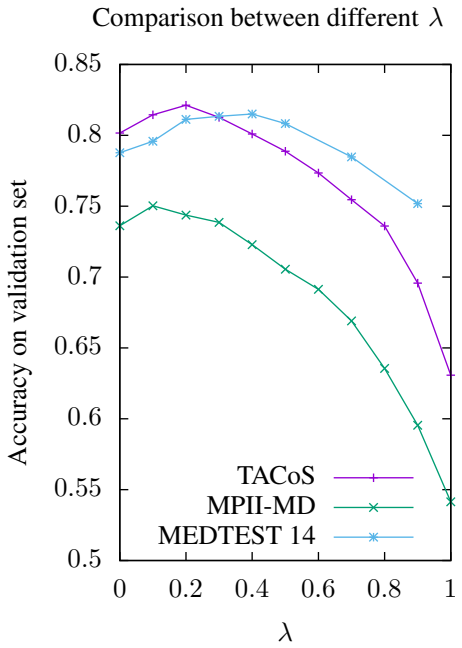

Fig. 9 The effectiveness of dual-channel learning to rank. We conduct experiments on the *Present-Easy* task to showcase. $\lambda = 0$ corresponds to using the sentence channel only and $\lambda = 1$ corresponds to using the word channel only

mance is shown in Fig. 9. As can be seen, it is beneficial to integrate word representations during training, and sentences are weighted more than words. This is because our visual features represent more global abstraction, which corresponds to sentence representation, whereas specific object features corresponding to word representation have not been considered in this work. We will explore this direction in detail in future works.

Biases in answers Jabri et al. (2016) found that there is a strong bias in Visual7W dataset (Zhu et al. 2016). $\lambda = 1$ in Fig. 9 indicates that there are biases in the answers. We report the results of $\lambda = 1$ in the *Present-Easy* task in Table 3 (*Visual + Answer-only*). We also conduct another experiment by introducing the question type prior where question type is known before answering the question. We answer the questions by selecting the most common candidate (measured by the answer frequency in the training set) of the question type. If the candidate does not belong to the question type, we will not choose it as the answer. We call it the *Question type Prior* baseline. The results show that *Question type Prior* baseline has high performance, however, the question type is a strong bias as it could be a lot easier after knowing what the question is asking about.

5.2 Evaluation of Inferring the Past and Predicting the Future

We first show the results of our GRU model in all tasks in Table 4. We illustrate some of the experiment results using our GRU model in Fig. 10 and show a number of wrong answers as well.

To show the effectiveness of our encoder–decoder approach in modeling video sequences, we compare our present model with a strong baseline—averaging frame-level features from GoogLeNet (Mean-GoogLeNet). We compare two representations by placing the visual input to our dual-channel ranking objective with the Mean-GoogLeNet or our GRU features. Note that the comparison is reasonable as both features have the same dimension of 1024 and we use the same transformation layer and same hyper-parameters during training. The results are shown in Table 5. From the results, we make the following observations:

(1) The GRU model outperforms Mean-GoogLeNet model in all cases, and performs relatively better than Mean-GoogLeNet in the tasks of inferring the past and predicting the future compared to describing the present. For the MPII-MD Easy dataset, the GRU model performs better than the Mean-GoogLeNet model by 2.2% in describing the present, and around 5.0% improvements are achieved for the past and future inferring. It shows the effectiveness of our GRU encoder–decoder framework in modeling temporal structures in videos. As our models are trained to reconstruct past and future sequences, they can represent the past and future in a more reasonable way than the Mean-GoogLeNet models. Our results also demonstrate the ability of our GRU model to

Table 4 Results of our GRU models on inferring past and predicting the future for TACoS and MPII-MD datasets

	TACoS				MPII-MD				MED			
	Split 1	Split 2	Split 3	Mean	Split 1	Split 2	Split 3	Mean	Split 1	Split 2	Split 3	Mean
Past-Easy	78.1	78.3	78.5	78.3	72.4	72.0	72.0	72.1	–	–	–	–
Past-Hard	65.8	64.4	63.9	64.7	47.0	47.0	46.9	47.0	–	–	–	–
Present-Easy	79.1	81.9	78.1	79.7	75.5	74.6	72.4	74.2	83.7	83.0	82.8	83.2
Present-Hard	66.9	66.2	68.2	67.1	47.4	49.0	48.3	48.2	63.0	63.9	62.3	63.1
Future-Easy	76.9	79.6	79.7	78.7	75.9	73.3	71.7	73.6	–	–	–	–
Future-Hard	66.1	65.8	69.9	67.3	47.1	48.8	48.1	48.0	–	–	–	–

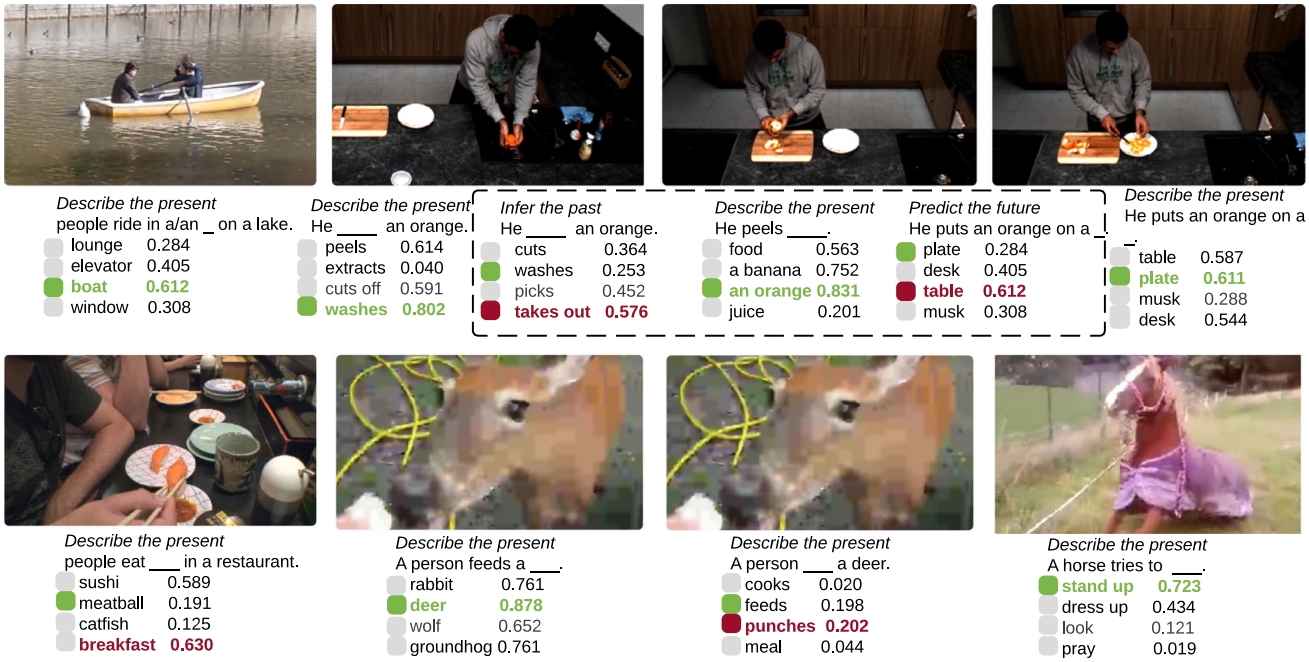

Fig. 10 Example results obtained from our model. Each candidate has a score corresponding to a clip. Correct answers are *marked in green* while failed cases are in *red* (Color figure online)

Table 5 Comparisons between ConvNets (Conv) and our model for past, present and future modeling

	Past			Present			Future		
	Conv	Ours	Improv	Conv	Ours	Improv	Conv	Ours	Improv
TACoS									
Easy	74.8	**78.3**	3.5	76.3	**79.7**	3.4	76.4	**78.7**	2.3
Hard	62.7	**64.7**	2.0	65.5	**67.1**	1.6	64.5	**67.3**	2.8
MPII-MD									
Easy	66.8	**72.1**	5.3	72.0	**74.2**	2.2	68.7	**73.6**	4.9
Hard	45.6	**47.0**	1.4	47.3	**48.2**	0.9	46.9	**48.0**	1.1

Relative improvements for each task are also listed
Bold values indicate the best performance

capture a wider range of temporal information than the Mean-GoogLeNet models. ConvNets trained from still frames can model temporal structures if the objects in a scene do not change too much in short intervals (one example would be in Fig. 1, where "cucumber" occurs in both the current clip and the future clip). However, in modeling longer sequences, ConvNets fail to make predictions due to lack of context.

(2) Our model can achieve better results for future prediction than for past inference. For future prediction, we feed the input frames in the order of 4, 5, 6 and the decoder is asked to reconstruct the frames in the order 7, 8, 9. We feed the same input for past inferring, but ask the decoder to reconstruct the target sequence of 1, 2, 3. As the future prediction model has shorter term dependencies than the past inferring model, it will be easier for the future prediction model to learn the temporal dependencies, which is consistent with the observations and hypothesis in Sutskever et al. (2014).

5.3 Limitations and Future Work

Although our results on question answering for video temporal context are encouraging, our model has multiple limitations. First, our model is only aware of context for at most 30 s (the longest unroll length). An alternative flexible and promising approach would be to incorporate the attention mechanism (Bahdanau et al. 2015) to learn longer sequences of context in videos. Additionally, our model sometimes fails to answer questions about detailed objects, due to lack of local visual features, i.e., region-level, bounding boxes based representation. We would like to integrate object detection ingredients to localize objects for better visual understanding. Lastly, we fixed the sentence and word representation learning in this work. Learning visual and language representations simultaneously remains an open problem, as indicated in Frome et al. (2013).

6 Conclusion

Unlike video captioning tasks which generate a generic and single description for a video clip, we introduced a temporal structure modeling approach for video question answering. We utilized an encoder–decoder model trained in an unsupervised way for visual context learning and propose a dual-channel learning-to-rank method to answer questions.

The proposed method is capable of modeling video temporal structure in a longer time range. We evaluated our approach on three datasets which have a large number of videos. The new approach outperforms the compared baselines, and achieves encouraging question answering results.

Acknowledgements Our work is partially supported by the Data to Decisions Cooperative Research Centre (www.d2dcrc.com.au), Google Faculty Award, and an Australian Government Research Training Program Scholarship. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the TITAN X (Pascal) GPU used for this research.

References

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., & Parikh, D. (2015). VQA: Visual question answering. In *International conference on computer vision (ICCV)*.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web* (pp. 722–735). Springer.

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *International conference on learning representations (ICLR)*.

Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2015). Learning phrase representations using RNN encoder—decoder for statistical machine translation. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Collobert, R., Kavukcuoglu, K., & Farabet, C. (2011). Torch7: A matlab-like environment for machine learning. In *Conference on neural information processing systems workshops (NIPS workshops)*.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Conference on computer vision and pattern recognition (CVPR)*.

Elliott, D., & Keller, F. (2014). Comparing automatic evaluation measures for image description. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., & Mikolov, T. (2013). DeViSE: A deep visual-semantic embedding model. In *Conference on neural information processing systems (NIPS)*.

Gan, C., Yang, Y., Zhu, L., Zhao, D., & Zhuang, Y. (2016). Recognizing an action using its name: A knowledge-based approach. *International Journal of Computer Vision (IJCV)*, *120*, 61–77.

Gao, H., Mao, J., Zhou, J., Huang, Z., Wang, L., & Xu, W. (2015). Are you talking to a machine? Dataset and methods for multilingual image question answering. In *Conference on neural information processing systems (NIPS)*.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Conference on computer vision and pattern recognition (CVPR)*.

Gong, Y., Ke, Q., Isard, M., & Lazebnik, S. (2014). A multi-view embedding space for modeling internet images, tags, and their semantics. *International Journal of Computer Vision (IJCV)*, *106*(2), 210–233.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hodosh, M., Young, P., & Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research (JAIR)*, *47*, 853–899.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning (ICML)*.

Jabri, A., Joulin, A., & van der Maaten, L. (2016). *Revisiting visual question answering baselines*. In European conference on computer vision (ECCV): Springer.

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Conference on computer vision and pattern recognition (CVPR)*.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In *Conference on neural information processing systems (NIPS)*.

Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Conference on neural information processing systems (NIPS)*.

Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A. C., & Berg, T. L. (2011). Baby talk: Understanding and generating image descriptions. In *Conference on computer vision and pattern recognition (CVPR)*.

Lebret, R., Pinheiro, P. O., & Collobert, R. (2015). Phrase-based image captioning. In *International conference on machine learning (ICML)*.

Lin, T.-Y., Maire, M., Belongie, S., Perona, P., Ramanan, D., Hays, J., et al. (2014). Microsoft COCO: Common objects in context. In *European conference on computer vision (ECCV)*.

Lin, X., & Parikh, D. (2015). Don't just listen, use your imagination: Leveraging visual common sense for non-visual tasks. In *Conference on computer vision and pattern recognition (CVPR)*.

Malinowski, M., & Fritz, M. (2014). A multi-world approach to question answering about real-world scenes based on uncertain input. In *Conference on neural information processing systems (NIPS)*.

Malinowski, M., Rohrbach, M., & Fritz, M. (2015). Ask your neurons: A neural-based approach to answering questions about images. In *International conference on computer vision (ICCV)*.

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A. L., & Murphy, K. (2015). Generation and comprehension of unambiguous object descriptions. In *Conference on computer vision and pattern recognition (CVPR)*.

MED. (2014). TRECVID MED 14. http://nist.gov/itl/iad/mig/med14.cfm.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Conference on neural information processing systems (NIPS)*.

Ordonez, V., Han, X., Kuznetsova, P., Kulkarni, G., Mitchell, M., Yamaguchi, K., et al. (2015). Large scale retrieval and generation of image descriptions. *International Journal of Computer Vision (IJCV)*, *119*, 46–59.

Pan, P., Xu, Z., Yang, Y., Wu, F., & Zhuang, Y. (2016). Hierarchical recurrent neural encoder for video representation with application to captioning. In *Conference on computer vision and pattern recognition (CVPR)*.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Regneri, M., Rohrbach, M., Wetzel, D., Thater, S., Schiele, B., & Pinkal, M. (2013). Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics (TACL)*, *1*, 25–36.

Ren, M., Kiros, R., & Zemel, R. (2015). Exploring models and data for image question answering. In *Conference on neural information processing systems (NIPS)*.

Rohrbach, A., Rohrbach, M., Tandon, N., & Schiele, B. (2015). A dataset for movie description. In *Conference on computer vision and pattern recognition (CVPR)*.

Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., & Schiele, B. (2013). Translating video content to natural language descriptions. In *International conference on computer vision (ICCV)*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, *115*(3), 211–252.

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Conference on neural information processing systems (NIPS)*.

Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015). Unsupervised learning of video representations using LSTMs. In *International conference on machine learning (ICML)*.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Conference on neural information processing systems (NIPS)*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Conference on computer vision and pattern recognition (CVPR)*.

Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., & Fidler, S. (2016). Movieqa: Understanding stories in movies through question-answering. In *Conference on computer vision and pattern recognition (CVPR)*. arXiv preprint arXiv:1512.02902.

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-RMSprop: Divide the gradient by a running average of its recent magnitude.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In *International conference on computer vision (ICCV)*.

Tu, K., Meng, M., Lee, M. W., Choe, T. E., & Zhu, S. C. (2014). Joint video and text parsing for understanding events and answering queries. *IEEE MultiMedia*, *21*(2), 42–70.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, *9*, 2579–2605.

Vedantam, R., Lawrence Zitnick, C., & Parikh, D. (2015). CIDEr: Consensus-based image description evaluation. In *Conference on computer vision and pattern recognition (CVPR)*.

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., & Saenko, K. (2015). Sequence to sequence—video to text. In *International conference on computer vision (ICCV)*.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Conference on computer vision and pattern recognition (CVPR)*.

Vondrick, C., Pirsiavash, H., & Torralba, A. (2015). Anticipating the future by watching unlabeled video. *Conference on computer vision and pattern recognition (CVPR)*.

Wang, H., Kläser, A., Schmid, C., & Liu, C. L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision (IJCV)*, *103*(1), 60–79.

Wu, Q., Wang, P., Shen, C., Dick, A., & van den Hengel, A. (2016). Ask me anything: Free-form visual question answering based on knowledge from external sources. In *Conference on computer vision and pattern recognition (CVPR)*.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., & Bengio, Y. (2015a). Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*.

Xu, Z., Yang, Y., & Hauptmann, A. G. (2015b). A discriminative CNN video representation for event detection. In *Conference on computer vision and pattern recognition (CVPR)*

Yan, Y., Nie, F., Li, W., Gao, C., Yang, Y., & Xu, D. (2016). Image classification by cross-media active learning with privileged information. *IEEE Transactions on Multimedia*, *18*(12), 2494–2502.

Yang, Y., Xu, D., Nie, F., Luo, J., & Zhuang, Y. (2009). Ranking with local regression and global alignment for cross media retrieval. In *Proceedings of the 17th ACM international conference on multimedia* (pp. 175–184). ACM.

Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., & Courville, A. (2015). Describing videos by exploiting temporal structure. In *International conference on computer vision (ICCV)*.

Young, P., Lai, A., Hodosh, M., & Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics (TACL)*, *2*, 67–78.

Yu, H., & Siskind, J. M. (2013). Grounded language learning from video described with sentences. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*.

Yu, L., Park, E., Berg, A. C., & Berg, T. L. (2015). Visual Madlibs: Fill in the blank image generation and question answering. In *International conference on computer vision (ICCV)*.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. arXiv preprint arXiv:1409.2329.

Zhu, Y., Groth, O., Bernstein, M., & Fei-Fei, L. (2016). Visual7w: Grounded question answering in images. In *Conference on computer vision and pattern recognition (CVPR)*.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *International conference on computer vision (ICCV)*.