MarioQA: Answering Questions by Watching Gameplay Videos

Jonghwan Mun* Paul Hongsuck Seo* Ilchae Jung Bohyung Han
Department of Computer Science and Engineering, POSTECH, Korea
{choco1916, hsseo, chey0313, bhhan}@postech.ac.kr

Abstract

We present a framework to analyze various aspects of models for video question answering (VideoQA) using customizable synthetic datasets, which are constructed automatically from gameplay videos. Our work is motivated by the fact that existing models are often tested only on datasets that require excessively high-level reasoning or mostly contain instances accessible through single frame inferences. Hence, it is difficult to measure capacity and flexibility of trained models, and existing techniques often rely on ad-hoc implementations of deep neural networks without clear insight into datasets and models. We are particularly interested in understanding temporal relationships between video events to solve VideoQA problems; this is because reasoning temporal dependency is one of the most distinct components in videos from images. To address this objective, we automatically generate a customized synthetic VideoQA dataset using Super Mario Bros. *gameplay videos so that it contains events with different levels of reasoning complexity. Using the dataset, we show that properly constructed datasets with events in various complexity levels are critical to learn effective models and improve overall performance.*

1. Introduction

While deep convolutional neural networks trained on large-scale datasets have been making significant progress on various visual recognition problems, most of these tasks focus on the recognition in the same or similar levels, *e.g.*, objects [16, 17], scenes [27], actions [15, 19], attributes [2, 26], face identities [21], etc. On the other hand, image question answering (ImageQA) [3, 4, 5, 6, 10, 14, 22, 23, 24, 25, 28] addresses a holistic image understanding problem, and handles diverse recognition tasks in a single framework. The main objective of ImageQA is to find an answer relevant to a pair of an input image and a question by capturing information in various semantic levels. This problem is often formulated with deep neural networks, and has been successfully investigated thanks to advance of representation learning techniques and release of outstanding pretrained deep neural network models [3, 6, 10, 23, 25].

Video question answering (VideoQA) is a more challenging task and is recently introduced as a natural extension of ImageQA in [13, 18, 29]. In VideoQA, it is possible to ask a wide range of questions about temporal relationship between events such as dynamics, sequence, and causality. However, there is only limited understanding about how effective VideoQA models in capturing various information from videos, which is partly because there is no proper framework including dataset to analyze models. In other words, it is not straightforward to identify main reason of failure in VideoQA problems—dataset vs. trained model.

There exist a few independent datasets for VideoQA [13, 18, 29], but they are not well-organized enough to estimate capacity and flexibility of models accurately. For example, answering questions in MovieQA dataset [18] requires too high-level understanding about movie contents, which is almost impossible to extract from visual cues only (*e.g.*, calling off one's tour, corrupt business, and vulnerable people) and consequently needs external information or additional modalities. On the other hand, questions in other datasets [13, 29] often rely on static or time-invariant information, which allows to find answers by observing a single frame with no consideration of temporal dependency.

To facilitate understanding of VideoQA models, we introduce a novel analysis framework, where we generate a customizable dataset using *Super Mario* video gameplays, referred to as *MarioQA*. Our dataset is automatically generated from gameplay videos and question templates to contain desired properties for analysis. We employ the proposed framework to analyze the impact of a properly constructed dataset to answering questions, where we are particularly interested in the questions related to temporal reasoning of events. The generated dataset consists of three subsets, each of which contains questions with a different level of difficulty in temporal reasoning. Note that, by controlling complexity of questions, individual models can be trained and evaluated on different subsets. Due to its synthetic nature, we can eliminate ambiguity in answers, which is often problematic

*Both authors contributed equally.

in existing datasets, and make evaluation more reliable.

Our contribution is three-fold as summarized below:

- We propose a novel framework for analyzing VideoQA models, where a customized dataset is automatically generated to have desired properties for target analysis.
- We generate a synthetic VideoQA dataset, referred to as MarioQA, using Super Mario gameplay videos with their logs and a set of predefined templates to understand temporal reasoning capability of models.
- We present the benefit of our framework to facilitate analysis of algorithms and show that a properly generated dataset is critical to performance improvement.

The rest of the paper is organized as follows. We first review related work in Section 2. Section 3 and 4 present our analysis framework with a new dataset, and discuss several baseline neural models, respectively. We analyze the models in Section 5, and conclude our paper in Section 6.

2. Related Work

2.1. Synthetic Testbeds for QA Models

The proposed method provides a framework for VideoQA model analysis. Likewise, there have been several attempts to build synthetic testbeds for analyzing QA models [7, 20]. For example, bAbI [20] constructs a testbed for textual QA analysis with multiple synthetic subtasks, each of which focuses on a single aspect in textual QA problem. The datasets are generated by simulating a virtual world given a set of actions by virtual actors and a set of constraints imposed on the actors. For ImageQA, CLEVR [7] dataset is recently released to understand visual reasoning capability of ImageQA models. It aims to analyze how well ImageQA models generalizes compositionality of languages. Images in CLEVR are synthetically generated by randomly sampling and rendering multiple predetermined objects and their relationships.

2.2. VideoQA Datasets

Zhu *et al.* [29] constructed a VideoQA dataset in three domains using existing videos with grounded descriptions, where the domains include cooking scenarios [11], movie clips [12] and web videos [1]. The fill-in-the-blank questions are automatically generated by omitting a phrase (a verb or noun phrase) from the grounded description and the omitted phrase becomes the answer. For evaluation, the task is formed as answering multiple choice questions with four answer candidates. Similarly, [13] introduces another VideoQA dataset with automatically generated fill-in-the-blank questions from LSMDC movie description dataset. Although the task for these datasets has a clear evaluation metric, the evaluation is still based on exact word matching rather than matching their semantics. Moreover, the questions can be answered by simply observing any single frame without need for temporal reasoning.

MovieQA dataset [18] is another public benchmark of VideoQA based on movie clips. This dataset contains additional information in other modalities including plot synopses, subtitles, DVS and scripts. The question and answer (QA) pairs are manually annotated based on the plot synopses without watching movies. The tasks in MovieQA dataset are difficult because the most questions are about the story of movies rather than about the visual contents of video clips. Hence, it needs to refer to external information other than the video clips, and not appropriate to evaluate trained models in terms of video understanding capability.

Contrary to these datasets, MarioQA is composed of videos with multiple events and event-centric questions. Data in MarioQA require video understanding over multiple frames for reasoning temporal relationship between events but do not need extra information to find answers.

2.3. Image and Video Question Answering

ImageQA Because ImageQA needs to handle two input modalities, *i.e*., image and question, [6] presents a method of fusing two modalities to obtain rich multi-modal representations. To handle the diversity of target tasks in ImageQA, [3] and [10] propose adaptive architecture design and parameter setting techniques, respectively. Since questions often refer to particular objects within input images, many networks are designed to attend to relevant regions only [3, 5, 14, 24, 25]. However, it is not straightforward to extend the attention models learned in ImageQA to VideoQA tasks due to additional temporal dimension in videos.

VideoQA In MovieQA [18], questions depend heavily on the textual information provided with movie clips, so models are interested in embedding the multi-modal inputs on a common space. Video features are obtained by simply average-pooling image features of multiple frames. On the other hand, [29] employs gated recurrent units (GRU) for sequential modeling of videos instead of simple pooling methods. In addition, unsupervised feature learning is performed to improve video representation power. However, none of these methods explore attention models although visual attention turns out to be effective in ImageQA [3, 5, 14, 24, 25].

3. VideoQA Analysis Framework

This section describes our VideoQA analysis framework for temporal reasoning capability using MarioQA dataset.

3.1. MarioQA Dataset Construction

MarioQA is a new VideoQA dataset in which videos are recorded from gameplays and questions are about the events

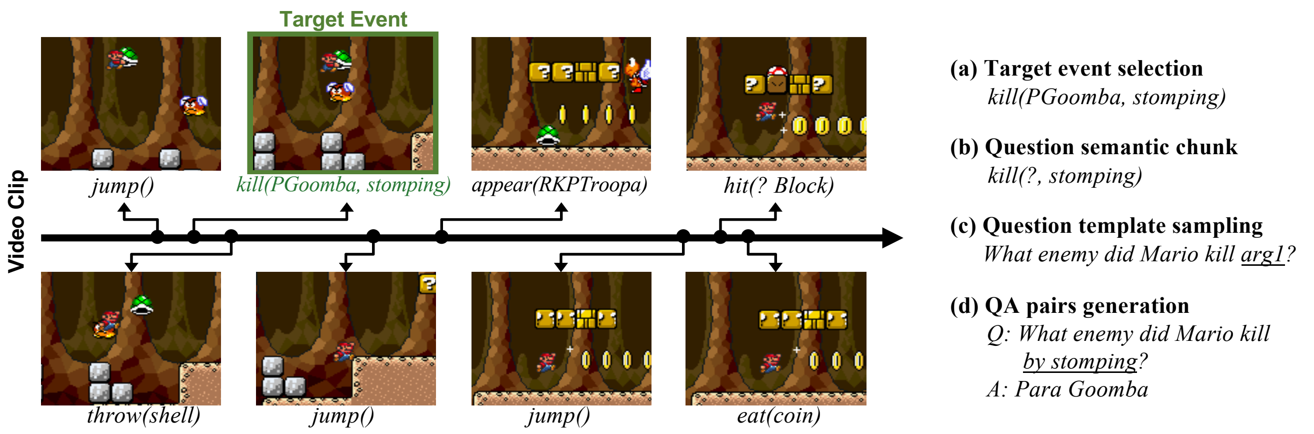


Figure 1: Overall QA generation procedure. Given a gameplay video and event logs shown on the left, (a) target event is selected (marked as a green box), (b) question semantic chunk is generated from the target event, (c) question template is sampled from template pool, and (d) QA pairs are generated by filling the template and the linguistically realizing answer.

occurring in the videos. We use the *Infinite Mario Bros.*[1] game, which is a variant of Super Mario Bros. with endless random level generation, to collect video clips with event logs and generate QA pairs automatically from extracted events using manually constructed templates. Our dataset mainly contains questions about temporal relationships of multiple events to analyze temporal reasoning capability of models. Each example consists of a 240×320 video clip containing multiple events and a question with corresponding answer. Figure 1 illustrates our data collection procedure.

Design principle We build the dataset based on two design principles to overcome the existing limitations. First, the dataset is aimed to verify model's temporal reasoning capability of events in videos. To focus on this main issue, we remove questions that require additional or external information to return correct answers and highlight model capacity for video understanding. Second, given a question, the answer should be clear and unique to ensure meaningful evaluation and interpretation. Uncertainty in answers, ambiguous linguistic structure of questions, and multiple correct answers may result in inconsistent or even wrong analysis, and make algorithms stuck in local optima.

Domain selection We choose gameplay videos as our video domain due to the following reasons. First, we can easily obtain a large amount of videos that contain multiple events with their temporal dependency. Second, learning complete semantics in gameplay videos is relatively easy compared to other domains due to their representation simplicity. Third, occurrence of an event is clear and there is no perceptual ambiguity. In real videos, answers for a question may be diverse depending on annotators because of subjective perception of visual information. On the contrary, we can simply access the oracle within the code to find answers.

[1]https://github.com/cflewis/Infinite-Mario-Bros

Target event and clip extraction We extract 11 distinct events $E = \{$kill, die, jump, hit, break, appear, shoot, throw, kick, hold, eat$\}$ with their arguments, *e.g.*, *agent*, *patient* and *instrument* from gameplays. For each extracted event as a target, we randomly sample video clips containing the target event with duration of 3 to 6 seconds. We then check the uniqueness of the target event within the sampled clip. For instance, the event *kill* with its arguments *PGoomba* and *stomping* is a unique target event among 8 extracted events in the video clip of Figure 1. This uniqueness check process rejects questions involving multiple answers since they cause ambiguity in evaluation.

Template-based question and answer generation Once the video clips of unique events are extracted, we generate QA pairs from the extracted events. We randomly eliminate one of the event arguments to form a *question semantic chunk* and generate a question from the question semantic chunk using predefined question templates. For example, an argument *PGoomba* is removed to form a question semantic chunk, *kill(?, stomping)*, from the event *kill(PGoomba, stomping)* in Figure 1. Then, a question template *'What enemy did Mario kill $\underline{\textit{arg1}}$?'* is selected from the template pool for the question semantic chunk. Finally, a question is generated by filling the template with a phrase *'by stomping'*, which linguistically realizes an argument, *stomping*. We use the template-based question generation because it allows to control level of semantics required for question answering. When annotators are told to freely create questions, it is hard to control the required level of semantics. So, we ask human annotators to create multiple linguistic realizations of a question semantic chunk. After question generation, a corresponding answer is also generated from the eliminated event argument, *i.e.*, *Para Goomba* in the above example. Note that the dataset is easily customized by updating QA templates to further reflect analytical perspectives and demands.

Table 1: Comparisons of the three VideoQA datasets.

Dataset	Video	Extra Info.	Goal	Domain	Data Source	Method	# of QA
Fill-in-the-blank [29]	✓	–	filling blanks of video descriptions	cooking scenario, movies, web videos	video description dataset	automatic generation	390,744
MovieQA [18]	✓	✓	answering questions for movie stories	movies	plot synopses, movies, subtitles, DVS, scripts	human annotation	15,000
MarioQA (ours)	✓	–	answering event-centric questions by temporal reasoning	gameplay (Infinite Mario Bros.)	gameplays, QA templates	automatic generation	187,757

Table 2: Examples of QA pairs in the three VideoQA datasets with answers boldfaced.

Fill-in-the-blank [29]	MovieQA [18]	MarioQA (ours)
• The man grab a ______. / **plate** • He slice the ______ very thinly, run through slice them twice. / **garlic** • The woman wash the plum off with ______ and the towel. / **water** • People eat and ______ at a resort bar / **drink** • A little ______ play hide and seek with a woman. / **girl** • a small group of guy climb ______ and boulder in a forest / **rock**	• What is Rob's passion in life? / **Music** • How many years does Lucas serve in prison? / **15 years** • How does Epps's wife demonstrate her attitude to Patsey? / **By constantly humiliating and degrading her** • Who eventually hires Minny? / **Celia Foote** • When was Milk assassinated? / **1978** • Does Solomon get the promised job? / **No**	• Where was the Green Koopa Troopa stomped? / **Hill** • Which enemy was killed by Mario's stomp after Mario hit a coin block? / **Goomba** • How many fireballs did Mario shoot? / **5** • How many times did Mario jump before holding a shell? / **3** • What is the type of stage? / **Cave** • What is the state of Mario when eating a mushroom? / **Fire form**

3.2. Characteristics of Dataset for Model Analysis

There are three question types in MarioQA to maintain diversity. The followings are examples of event-centric, counting and state questions, respectively: *'What did Mario hit before killing Goomba?'*, *'How many coins did Mario eat after a Red Koopa Paratroopa appears?'* and *'What was Mario's state when Green Koopa Paratroopa appeared?'* While questions in the three types generally require observation over multiple frames to find answers, a majority of state questions just need a single frame observation about objects and/or scene due to the uniqueness of the state within a clip.

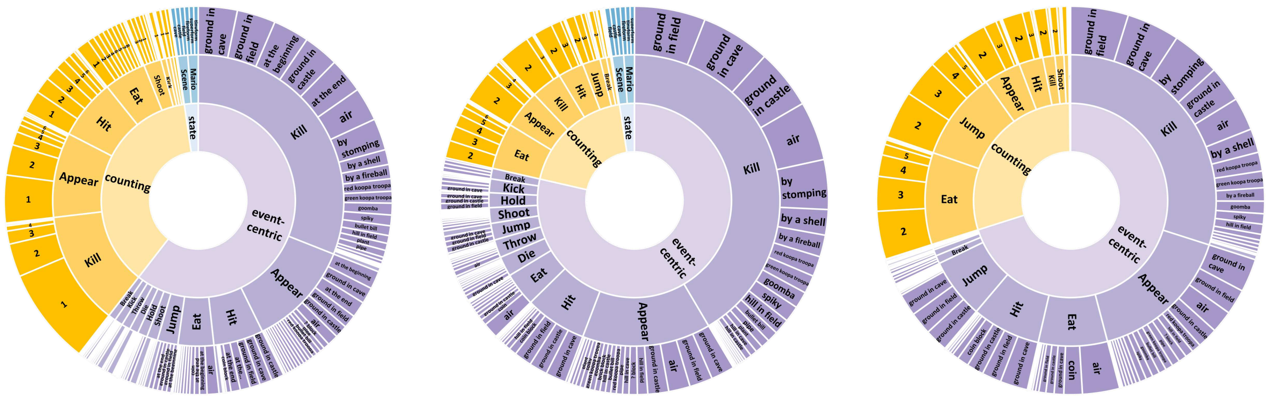
As seen in the above examples, multiple events in a single video may be temporally related to each other, and understanding such temporal dependency is an important aspect in VideoQA. In spite of importance of this temporal dependency issue in videos, it has not been explored explicitly due to lack of proper datasets and complexity of tasks. Thanks to the synthetic property, we can generate questions about temporal relationships conveniently.

We construct MarioQA dataset with three subsets, which contain questions with different characteristics in temporal relationships: questions with no temporal relationship (NT), with easy temporal relationship (ET) and with hard temporal relationships (HT). NT asks questions about unique events in the entire video without any temporal relationship phrase. ET and HT have questions with temporal relationships in different levels of difficulty. While ET contains questions about globally unique events, HT involves distracting events making a VQA system choose a right answer out of multiple identical events using temporal reasoning; for a target event *kill(PGoomba, stomping)*, any *kill(*, *)* events in the same video clip are considered as distracting events. Note that the answer of a question *'How many times did Mario jump after throwing a shell?'* about the video clip in Figure 1 is not 3 but 2 due to its temporal constraint. Note that the generated questions are still categorized into one of three types—event-centric, counting and state questions.

3.3. Dataset Statistics

From a total of 13 hours of gameplays, we collect 187,757 examples with automatically generated QA pairs. There are 92,874 unique QA pairs and each video clip contains 11.3 events in average. There are 78,297, 64,619 and 44,841 examples in NT, ET and HT, respectively. Note that there are 3.5K examples that can be answered using a single frame of video; the portion of such examples is only less than 2%. The other examples are event-centric; 98K examples require to focus on a single event out of multiple ones while 86K need to recognize multiple events for counting (55K) or identifying their temporal relationships (44K). Note that there are instances that belong to both cases.

Some types of events are more frequently observed than others due to the characteristics of the game, which is also common in real datasets. To make our dataset more balanced, we have a limit for the maximum number of same QA pairs. The QA pair distribution of each subset is depicted in Figure 2. The innermost circles show the distributions of the three question types. The portion of event-centric questions is much larger than those of the other types in all three subsets as we focus on the event-centric questions. The middle circles present instance distributions in each question type, where we observe a large portion of *kill* event since *kill*



(a) NT (b) ET (c) HT

Figure 2: The distributions of question and answer pairs in three subsets of MarioQA.

events occur with more diverse arguments such as multiple kinds of enemies and weapons. The outermost circles show the answer distributions related to individual events or states.

The characteristics of VideoQA datasets are presented in Table 1. The number of examples in [29] is larger than the other datasets but fragmented into many subsets, which are hard to be used as a whole. MovieQA [18] dataset has extra information in other modalities. Both datasets have limitations in evaluating model capacity for video understanding as in the examples in Table 2. The questions in [29] are mainly about the salient contents throughout the videos. These questions can be answered by understanding a single frame rather than multiple ones. On the other hand, the questions in MovieQA are often too difficult to answer by watching videos as they require very high-level abstraction about movie story. In contrast, MarioQA contains videos with multiple events and their temporal relationships. The event-centric questions with temporal dependency allow us to evaluate whether the model can reason temporal relationships between multiple events.

4. Neural Models for MarioQA

We describe our neural baseline models for MarioQA. All networks comprise of three components: question embedding, video embedding and classification networks depicted in Figure 3. We explore each component in detail below.

4.1. Question Embedding Network

Recurrent neural networks (RNN) with memory units such as long short term memory (LSTM) or gated recurrent unit (GRU) are widely used in ImageQA [4, 6, 10, 22, 23]. Following [10] and [29], we obtain a question embedding vector using the pretrained GRU on the large corpus [9]. The question embedding vector $f^q \in \mathbb{R}^{2400}$ is given by

$$f^q = \mathrm{SkipThought}(q) \tag{1}$$

where $\mathrm{SkipThought}(\cdot)$ denotes a pretrained question embedding network and q is an input question.

4.2. Video Embedding Network

We employ a 3D fully convolutional network (3DFCN) for video feature extraction. The 3DFCN is composed of five $3 \times 3 \times 3$ convolution layers with four pooling layers between them. The first two pooling layers only pool features in the spatial dimensions but not in the temporal dimension whereas the last two pooling layers pool in all spatio-temporal dimensions. We rescale input videos to 120×160 and sample K key frames with temporal stride 4 before feeding to the network. The output of the 3DFCN is a feature volume $\boldsymbol{f}^v \in \mathbb{R}^{512\times T\times 7\times 10}$, which is given by

$$\boldsymbol{f}^v = \mathrm{3DFCN}(V), \tag{2}$$

where $T = K/4$ and V denotes an input video. Once the video features are extracted from the 3DFCN, we embed these features volume $\boldsymbol{f}^v$ onto a low-dimensional spaces using one of the following ways.

4.2.1 Embedding with Temporal Attention

As many events in a video from MarioQA are not relevant to input question, temporal localization of target events in videos may be important to answer questions. The temporal attention model embeds the entire video feature volume onto a low-dimensional space by interpolating features in the temporal dimension based on the weights learned for attention. In this model, a frame feature is first extracted through an average pooling from the feature map at every location in the temporal dimension. Formally, a frame feature f_t^{frame} at time t is obtained from a video feature map $\boldsymbol{f}_t^v$ by

$$f_t^{\text{frame}} = \underset{i,j}{\mathrm{avgpool}}(\boldsymbol{f}_t^v), \tag{3}$$

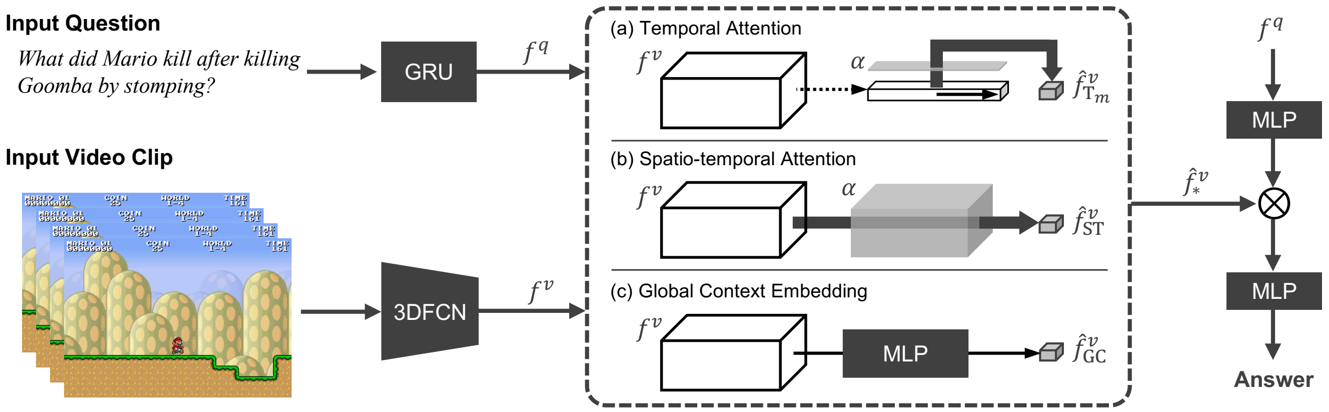

Input Question
What did Mario kill after killing Goomba by stomping?
Input Video Clip
GRU
3DFCN
f^q
f^v
(a) Temporal Attention
α
$\hat{f}^v_{\mathrm{T}_m}$
(b) Spatio-temporal Attention
$\hat{f}^v_{\mathrm{ST}}$
(c) Global Context Embedding
MLP
$\hat{f}^v_{\mathrm{GC}}$
$\hat{f}^v_*$
Answer

Figure 3: Architecture of the neural models in our analysis. The encoders—3DFCN and GRU for video feature extraction and question embedding, respectively—as well as the final classification network are shared by all neural models. We introduce neural baseline models with three different attention mechanisms on MarioQA: (a) temporal attention, (b) spatio-temporal attention and (c) global context embedding with no attention.

where $\mathrm{avgpool}_{i,j}$ denotes the average pooling operation in spatial dimensions and the resulting frame features are used for temporal attention process.

Single-step temporal attention Given a frame feature f_t^{frame} and a question embedding f^q, the temporal attention probability α_t^1 for each frame feature is obtained by

$$s_t^1 = \mathrm{att}(f_t^{\text{frame}}, f^q), \tag{4}$$
$$\alpha_t^1 = \mathrm{softmax}_t(\boldsymbol{s}^1) \tag{5}$$

where $\mathrm{att}(\cdot,\cdot)$ is a multilayer perceptron composed of one hidden layer with 512 nodes and a scalar output. The temporally attended video embedding $\hat{f}^v_{\mathrm{T}_1}$ is obtained using the attention probabilities as

$$\hat{f}^v_{\mathrm{T}_1} = \sum_t^T \alpha_t^1 f_t^{\text{frame}}. \tag{6}$$

Multi-step temporal attention We also employ multi-step temporal attention models, which applies the temporal attention multiple times. We follow the multiple attention process introduced in [25]. The temporally attended video embedding with m steps denoted by $\hat{f}^v_{\mathrm{T}_m}$ is obtained by the same process except that the question embedding is refined by adding the previous attended embedding $\hat{f}^v_{\mathrm{T}_{m-1}}$ as

$$s_t^m = \mathrm{att}(f_t^{\text{frame}}, f^q + \hat{f}^v_{\mathrm{T}_{m-1}}), \tag{7}$$
$$\alpha_t^m = \mathrm{softmax}_t(\boldsymbol{s}^m), \tag{8}$$

where $\hat{f}^v_{\mathrm{T}_0}$ is a zero vector. The temporally attended video embedding $\hat{f}^v_{\mathrm{T}_m}$ with m steps is given by a similar linear combination in Eq. (6).

4.2.2 Embedding with Spatio-Temporal Attention

We can attend to a single feature in a spatio-temporal feature volume. The attention score $s_{t,i,j}$ for each feature $f^v_{t,i,j}$ and the attention probability $\alpha_{t,i,j}$ is given respectively by

$$s_{t,i,j} = \mathrm{att}(f^v_{t,i,j}, f^q) \quad \text{and} \tag{9}$$
$$\alpha_{t,i,j} = \mathrm{softmax}_{t,i,j}(\boldsymbol{s}), \tag{10}$$

where the softmax function is applied throughout the spatio-temporal space to normalize the attention probabilities. The spatio-temporally attended video embedding $\hat{f}^v_{\mathrm{ST}}$ is then obtained by

$$\hat{f}^v_{\mathrm{ST}} = \sum_t \sum_i \sum_j \alpha_{t,i,j} f^v_{t,i,j}. \tag{11}$$

4.2.3 Global Context Embedding

A popular video embedding method is to embed the entire video feature volume using fully-connected layers [8, 19]. This method is more appropriate to capture the global context than the attention-based video embedding models described above. We build a multi-layer perceptron that has two fully-connected layers with 512 hidden nodes and 512 dimensional output layer for video embedding. That is, the video embedding $\hat{f}^v_{\mathrm{GC}}$ is obtained by

$$\hat{f}^v_{\mathrm{GC}} = \mathrm{vemb}(\boldsymbol{f}^v_{\text{flat}}) \tag{12}$$

where vemb is the MLP for the video embedding and $\boldsymbol{f}^v_{\text{flat}}$ is the flattened video feature volume.

4.3. Classification Network

The classification network predicts the final answer given a video embedding $\hat{f}^v_*$ and a question embedding f^q, where

Table 3: Accuracies [%] for the models on test splits. Models are trained on different combinations of subsets: NT, NT+ET and NT+ET+HT to see the impact of each subset on accuracies. Each trained model is then tested on test split of each subset.

	Trained on NT (case 1)				Trained on NT+ET (case 2)				Trained on NT+ET+HT (case 3)			
	NT	ET	HT	ALL	NT	ET	HT	ALL	NT	ET	HT	ALL
V	21.29	32.33	31.36	27.49	20.67	35.80	34.25	29.12	21.16	35.32	34.00	29.10
Q	40.86	24.65	26.50	31.85	40.23	35.52	36.46	37.71	39.79	35.67	39.65	38.34
AP	56.76	33.14	29.39	42.09	58.79	62.57	59.02	60.15	60.03	66.49	64.95	63.43
1-T	56.65	42.99	41.34	48.29	61.61	64.21	60.20	62.17	64.28	69.64	67.21	66.82
2-T	53.96	39.32	40.71	45.76	62.87	66.42	61.75	63.82	64.05	66.13	65.65	65.15
ST	60.18	47.79	45.87	52.50	62.80	68.44	61.87	64.52	66.38	72.73	69.27	69.26
GC	55.83	29.89	25.24	39.60	65.62	74.27	61.35	67.58	66.47	75.10	68.89	70.02

Table 4: Number of examples in each split of subsets. The training, validation and test splits in each subset are obtained by random sampling of 60%, 20%, 20% of data from each subset.

	train	valid	test	total
NT	46,978	15,660	15,659	78,297
ET	38,771	12,924	12,924	64,619
HT	26,904	8,969	8,968	44,841

$* \in \{\mathrm{T}_1, \mathrm{T}_m, \mathrm{ST}, \mathrm{GC}\}$. The question embedding f^q is first transformed to the common space with the video embedding $\hat{f}^v$, and the transformed embedding is given by

$$\hat{f}^q = \sigma(W_q f^q), \quad (13)$$

where W_q is 512×2400 weight matrix and σ is a nonlinear function such as ReLU. Then, we fuse the two embedded vectors and generate the final classification score by

$$S = \mathrm{softmax}(W_{\mathrm{cls}}(\hat{f}^v \odot \hat{f}^q)), \quad (14)$$

where W_{cls} is the weight matrix for final classification layer.

5. Experiments

5.1. Experimental Setting

We have three subsets in MarioQA dataset as presented in Table 4. We aim to analyze the impact of questions with temporal relationships in training, so we train models on the following three combinations of the subsets: NT (case 1), NT+ET (case 2) and NT+ET+HT (case 3). Then, these models are evaluated to verify temporal reasoning capability on the test split of each subset. We implement two versions of the temporal attention models with one and two attention steps (1-T and 2-T) following [25]. The spatio-temporal attention model (ST) and the global context embedding (GC) are also implemented. In addition to these models, we build three simple baselines:

- **Video Only (V)** Given a video, the model predicts an answer without knowing the question. We perform video embedding by Eq. (12) and predict answers using a multi-layer perceptron with one hidden layer.
- **Question Only (Q)** This model predicts an answer by observing questions but without seeing videos. The same question embedding network is used with the classification.
- **Average Pooling (AP)** This model embeds the video feature volume $\boldsymbol{f}^v$ by average pooling throughout the spatio-temporal space and use it for final classification. This model is for comparisons with the attention models as the average pooling is equivalent to assigning the uniform attention to every spatio-temporal location.

All the models are trained end-to-end by the standard back-propagation from scratch while the question embedding network is initialized with a pretrained model [9]. The vocabulary sizes of questions are 136, 168 and 168 for NT, ET and HT, respectively, and the number of answer classes is 57.

In our scenario, the initial model of each algorithm is trained with NT only and we evaluate algorithms in all three subsets by simply computing the ratio of correct answers to the total number of questions. Then, we add ET and HT to training data one by one, and perform the same evaluation and observe tendency of performance change.

5.2. Results

Analysis on neural models Table 3 presents the overall results of our experiments. Obviously, two simple baselines (V and Q) show significantly lower performance than the others[2]. Although three attention-based models and AP outperform GC in case 1, GC becomes very competitive in case 2 and case 3. It is probably because network architectures with general capabilities such as fully-connected layers are more powerful than the linear combinations of attentive features; if the dataset is properly constructed involving examples with temporal relationships, GC is likely to achieve high performance. However, attention models may be able to gain more benefit from pretrained models, and GC is a more preferable model for our environment with short video clips.

Analysis on dataset variation Our results strongly suggest that proper training data construction would help to learn a better model. Figure 4 presents qualitative results for an HT question in all three cases. It shows that the models tend to predict the correct answer better as ET and HT are

[2]To demonstrate the strength of trained models, we evaluate random guess accuracies, which are 16.96, 12.66 and 12.66 (%) for NT, ET and HT, respectively. Also, The accuracies obtained by selecting the most frequent answer are 37.85, 32.29 and 39.00 (%) for NT, ET and HT, respectively.

Q: What enemy came in before a Red Koopa Troopa appears?

A: *Spiky*

	1-T	2-T	ST	GC
case 1	Ground in field	? block	Goomba	Ground in field
case 2	Goomba	Bullet bill	Spiky	Spiky
case 3	Spiky	Spiky	Spiky	Spiky

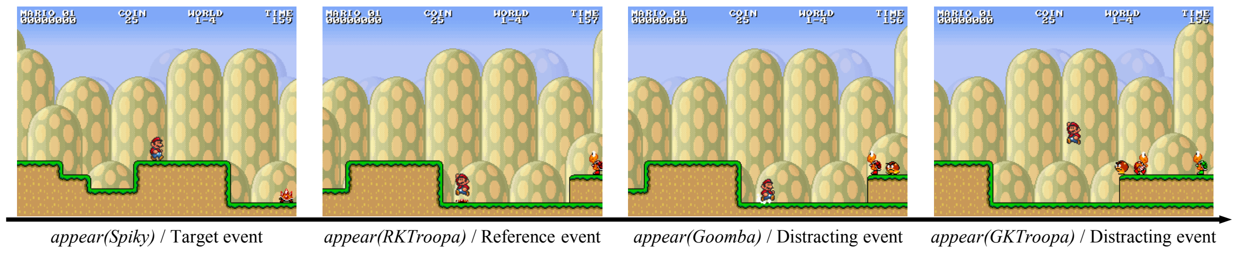

appear(Spiky) / Target event
appear(RKTroopa) / Reference event
appear(Goomba) / Distracting event
appear(GKTroopa) / Distracting event

Figure 4: Qualitative results of an HT question. Four frames of a video clip representing target, reference and two distracting events are presented with their events, where reference event means temporally related event to target event in question. Models are trained on different combinations of subsets: NT (case 1), NT+ET (case 2) and NT+ET+HT (case 3). Note that models tend to predict correct answers when trained on properly constructed training dataset while most models generate answers far from correct ones in case 1. Correct and incorrect answers are marked as green and red respectively while incorrect answers from distracting events are marked as blue.

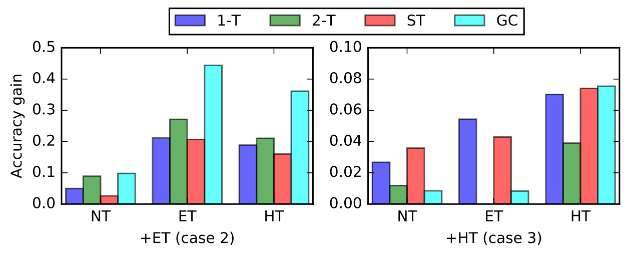


Figure 5: Accuracy gain of every model on each subset whenever a subset is added to training set. Added subsets are shown below each graph and each bar in graphs shows gain from accuracy without added subset. Test subsets are shown on x-axis.

Table 5: Results from GC in the three subsets with roughly the same number of training examples.

Train set	# of train example	NT	ET	HT	ALL
NT (case 1)	46,978	55.83	29.89	25.24	39.60
NT+ET (case 2)	46,990	53.39	58.54	60.16	56.78
NT+ET+HT (case 3)	46,970	54.49	60.83	61.50	58.35

added to training dataset. The quantitative impact of adding ET and HT to training data is illustrated in Figure 5. By adding ET to training dataset, we observe improvement of all algorithms with attention models (and AP) in all three subsets, where performance gains in ET are most significant consistently in all algorithms. The similar observation is found when HT is additionally included in the training dataset although the magnitudes of improvement are relatively small. This makes sense because the accuracies are getting more saturated as more data are used for training.

It is noticeable that training with the subsets that require more difficult temporal reasoning also improves performance of the subsets with easier temporal reasoning; training with ET or HT improves performance not only on ET and HT but also on NT. It is also interesting that training with ET still improves the accuracy on HT. Since ET does not contain any distracting events, questions in ET can be answered conceptually regardless of temporal relationships. However, the improvement on HT in case 2 intimates that the networks still learn a way of temporally relating events using ET.

One may argue that the improvement mainly comes from the increased number of training examples in case 2 and case 3. To clarify this issue, we train GC models for case 2 and case 3 using roughly the same number of training examples with case 1 (Table 5). Due to smaller training datasets, the overall accuracies are not as good as our previous experiment but the performance improvement tendency is almost same. It is interesting that three cases on NT testing set achieve almost same accuracy in this experiment even with less training examples in NT for case 2 and case 3. This fact shows that ET and HT are helpful to solve questions in NT.

6. Conclusion

We propose a new analysis framework for VideoQA and construct a customizable synthetic dataset, MarioQA. Unlike existing datasets, MarioQA focuses on event-centric questions with temporal relationships to evaluate temporal reasoning capability of algorithms. The questions and answers in MarioQA are automatically generated based on manually constructed question templates. We use our dataset for the analysis on the impact of questions with temporal relationships in training and show that properly collected dataset is critical to improve quality of VideoQA models. However, we believe that MarioQA can be used for further analyses on other perspectives of VideoQA by customizing the dataset with preferred characteristics in its generation.

Acknowledgement This work was partly supported by the ICT R&D program of MSIP/IITP [2014-0-00059 and 2016-0-00563] and the NRF grant [NRF-2011-0031648] in Korea. Samsung funded our preliminary work in part.

References

[1] Trecvid med 14. http://nist.gov/itl/iad/mig/med14.cfm, , 2014. 2

[2] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia. Multi-task cnn model for attribute prediction. *IEEE Transactions on Multimedia*, 2015. 1

[3] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *NAACL*, 2016. 1, 2

[4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 1, 5

[5] K. Chen, J. Wang, L.-C. Chen, H. Gao, W. Xu, and R. Nevatia. Abc-cnn: An attention based convolutional neural network for visual question answering. *arXiv preprint arXiv:1511.05960*, 2015. 1, 2

[6] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016. 1, 2, 5

[7] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 2

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 6

[9] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *NIPS*, 2015. 5, 7

[10] H. Noh, P. H. Seo, and B. Han. Image question answering using convolutional neural network with dynamic parameter prediction. In *CVPR*, 2016. 1, 2, 5

[11] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal. Grounding action descriptions in videos. *TACL*, 2013. 2

[12] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *CVPR*, 2015. 2

[13] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie description. *IJCV*, 2017. 1, 2

[14] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. In *CVPR*, 2016. 1, 2

[15] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 1

[18] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 2016. 1, 2, 4, 5

[19] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1, 6

[20] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015. 2

[21] L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 1

[22] Q. Wu, P. Wang, C. Shen, A. v. d. Hengel, and A. Dick. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *CVPR*, 2016. 1, 5

[23] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, 2016. 1, 5

[24] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*, 2016. 1, 2

[25] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016. 1, 2, 6, 7

[26] Y. Zhong, J. Sullivan, and H. Li. Face attribute prediction using off-the-shelf cnn features. In *International Conference on Biometrics (ICB)*, 2016. 1

[27] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 1

[28] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*, 2015. 1

[29] L. Zhu, Z. Xu, Y. Yang, and A. G. Hauptmann. Uncovering temporal context for video question and answering. *arXiv preprint arXiv:1511.04670*, 2015. 1, 2, 4, 5