# Where did I leave my keys? — Episodic-Memory-Based Question Answering on Egocentric Videos

Leonard Bärmann, Alex Waibel
Interactive Systems Lab, Karlsruhe Institute of Technology, Germany
{baermann,waibel}@kit.edu

## Abstract

*Humans have a remarkable ability to organize, compress and retrieve episodic memories throughout their daily life. Current AI systems, however, lack comparable capabilities as they are mostly constrained to an analysis with access to the raw input sequence, assuming an unlimited amount of data storage which is not feasible in realistic deployment scenarios. For instance, existing Video Question Answering (VideoQA) models typically reason over the video while already being aware of the question, thus requiring to store the complete video in case the question is not known in advance.*

*In this paper, we address this challenge with three main contributions: First, we propose the Episodic Memory Question Answering (EMQA) task as a specialization of VideoQA. Specifically, EMQA models are constrained to keep only a constant-sized representation of the video input, thus automatically limiting the computation requirements at query time. Second, we introduce a new egocentric VideoQA dataset called* QAEGO4D*, far larger than existing egocentric VideoQA datasets and featuring video length unprecedented in VideoQA datasets in general. Third, we present extensive experiments on the new dataset, comparing various baseline models in both the VideoQA and the EMQA setting. To facilitate future research on egocentric VideoQA as well as episodic memory representation and retrieval, we publish our code and dataset.*

## 1. Introduction

During our daily life, we humans collect a vast amount of experiences, which we intuitively filter, organize, aggregate and store in our episodic memory (EM) [34]. Although we certainly do not remember everything, we have the impressive capability to recall *relevant* information with a high precision, *e.g.*, when answering questions about our experiences in conversations with other people. Emulating such *episodic memory question answering* (EMQA) competence in artificial intelligence (AI) systems is desirable both for human assistance [9] as well as in robotics [2]. Most current AI systems, however, lack key components of these capabilities, even though first perceptual systems that attempted to build knowledge for use in a 24/7 always-on environment were already proposed in [13, 36]. When using existing Video Question Answering (VideoQA) models like [14, 38, 39] for EMQA, access to the complete video of the corresponding episode is required. Crucially, regardless of feature extraction or compression methods used, the storage required to process an input video grows linearly with the video length, thereby also implying linearly scaling computation costs at query time. While this is not problematic for current VideoQA datasets like [37, 38, 40] due to their short video lengths ranging from a few seconds to a few minutes, it prohibits future realistic deployment scenarios involving long-horizon episodic data, *e.g.*, a memorization helper device recording egocentric video in a 24/7-manner, or a robot equipped with an EM. Memory-Augmented Neural Networks (MANNs) [10, 45] solve this problem by processing the video into a fixed size memory and then answering questions only based on that representation.

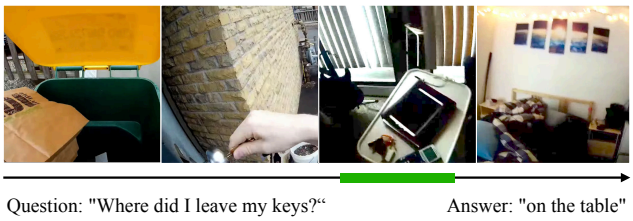


Figure 1. We introduce QAEGO4D, a new VideoQA dataset building on Ego4D [9] featuring long egocentric videos, natural language questions and answers as well as temporal window annotations.

In this paper, we address these issues by precisely defining the EMQA task, collecting a benchmark dataset based on Ego4D [9], as well as providing baseline experiments for both (unconstrained) VideoQA and the EMQA setting, where models are constrained to a constant-size memory.

Specifically, we present three major contributions:

As a first step, we define the task of EMQA as a specialization of VideoQA . In analogy to [45], the key difference lies in the allowed memory usage with respect to the input video length, which is unconstrained (usually linearly growing) in conventional VideoQA systems and bounded to a maximum, constant size in the EMQA setting. This has two crucial implications: First, it shifts the point in time when "relevant" content needs to be extracted to the time the video is given to the model (instead of the question time), thus, turning an off-line analysis into an on-line algorithm. Second, the upper bound on memory storage implies an upper bound of computation at query time, thereby, theoretically allowing the system to be used in a life-long manner. In VideoQA terminology, this means, it is possible to answer questions to arbitrarily long videos. Moreover, we phrase VideoQA and EMQA as open-ended, generative QA problems, hence, making the model more useful for realistic scenarios as it is not restricted to choosing from a set of predefined answers [4].

Our second contribution is to utilize the Ego4D dataset [9] to construct a new VideoQA dataset which we call QAEGO4D (see Fig. 1), especially suitable for the EMQA task. Ego4D is a dataset offering a huge amount of real-world egocentric video recordings. One of the challenges proposed by [9] along with Ego4D is called *Episodic Memory – Natural Language Queries* (EM – NLQ), where a a video and a corresponding question are given and the goal is to find the temporal window in the video which visually shows the answer to the question. To utilize this for EMQA and VideoQA in general, we employed human annotators to collect the corresponding natural language answers to the NLQ questions. This can be done efficiently, since the annotators can rely on the ground-truth temporal window, *i.e.*, they only have to watch video clips of a few seconds to find the correct response for each question.

Finally, our third contribution is to present extensive experiments on the new QAEGO4D dataset, applying a variety of baseline systems both on the VideoQA as well as the EMQA task. For VideoQA , we compare two recent systems from the literature, namely JustAsk [38] and Hierarchical Conditional Relation Networks (HCRN) [19], to a simple Transformer model [35] baseline and the Longformer [1] architecture. In the EMQA setting, we present results for a long-term variant of the Compressive Transformer [28], the Differentiable Neural Computer (DNC) [10], the Self-attentive-Associative-Memory-based Two-memory Model (STM) [18], as well as the Rehearsal Memory model [45]. Our experiments on the new dataset indicate the huge challenges QAEGO4D sets for future work. To facilitate such research, we publish[1] both our codebase as well as the QAEGO4D dataset.

[1] https://github.com/lbaermann/qaego4d

| Dataset | $\varnothing\|v\|$ | #V | #Q | Type | $\tau$? | $C$? |
|---|---|---|---|---|---|---|
| MovieQA [33] | 202.7 | 6,771 | 6,462 | movies | ✓ | |
| TVQA [20] | 76.2 | 21,793 | 152,545 | series | ✓ | |
| Act.Net-QA [40] | 180.0 | 5,800 | 58,000 | YouTube | | |
| iVQA [38] | 18.6 | 10,000 | 10,000 | YouTube | | ✓ |
| LifeQA [3] | 74.0 | 275 | 2,326 | YouTube | | |
| Pano-AVQA [41] | 5.1 | 5,400 | 51,700 | 360° | | |
| EgoVQA [6] | 2.2 | 520 | 520 | Ego | | |
| **QAEGO4D** | 495.1 | 1,325 | 14,513 | Ego | ✓ | ✓ |

Table 1. Comparison of related VideoQA datasets. $\varnothing|v|$ = average video length in seconds. #V = number of videos. #Q = number of questions. $\tau$? = target moment annotations present?. $C$? = answer confidence annotations present?. Ego = Egocentric videos

## 2. Related Work

**VideoQA** is a major topic in the vision-and-language research community. There exist plenty of VideoQA datasets [3, 20, 33, 37, 38, 40], of which Tab. 1 compares several ones related to this work. However, all of these have several limitations when applying them to the EMQA task: First, the typical video duration is rather short (3.3 minutes or less), thus, not demanding for an intermediate, limited-size storage in form of an EM. Second, except for EgoVQA [6] and Pano-AVQA [41], they show third-person video hence not transferring easily to realistic egocentric application scenarios [2, 9]. While EgoVQA features first-person videos, it is very limited in its size and video length. In contrast, our new QAEGO4D dataset contains a large amount of long egocentric videos (8 minutes on average). Furthermore, in contrast to previous work, QAEGO4D provides both target moment annotations as well as answer confidence estimations, which can both serve as an additional source of (weak) supervision.

Fostered by the amount of VideoQA datasets, there are also plenty of methods applied to VideoQA. Among others, recent models like SiaSamRea [39] and M3DC [25] reach state-of-the-art results on various VideoQA datasets by applying complex multimodal reasoning modules on top of common transformer and convolutional backbone architectures. JustAsk [38] learns open-ended VideoQA using a contrastive loss in combination with a huge set of automatically generated pretraining data. For a more thorough review of the broad VideoQA field, we refer the reader to recent surveys [27, 32]. Furthermore, many methods allow for flexibility concerning the used modalities, *e.g.*, M3DC uses audio stream, while HRCN [19] allows providing a textual stream of subtitles along with the video. In this work, we focus on two input modalities (video stream + question text) only, but the QAEGO4D dataset provides audio for two-thirds of the videos thus allowing for future extensions.

**Natural Language Video Localization** (NLVL) is closely related to VideoQA as the input is a natural language text and a video, and the model is supposed to find the most relevant temporal window in the video. As for VideoQA , there is a multitude of NLVL datasets, including ActivityNet Captions [16], Charades-STA [8], TACoS [30] and, most recently, the NLQ subset of Ego4D [9]. Recent state-of-the-art methods [24, 42, 43] propose several models highly specialized for solving the NLVL problem. In this work, we focus on evaluating VideoQA and EMQA performance, however we use the NLVL annotations as an additional source of supervision for our models.

**Memory-Based Methods** have a long tradition in neural networks research. Recurrent networks like LSTM [12] or GRU [5] are the simplest form, but have severe limitations regarding the capacity of the memory. More advanced memory-based networks like the Differential Neural Computer (DNC) [10] offer a much larger memory as well as a more flexible way of reading and writing the stored content. For instance, the Self-attentive-Associative-Memory-based Two-memory Model (STM) [18] splits responsibility for storing items and relations between these items into two separate memories. However, exploration of memory-based architectures featuring a limited storage requirement independent of input length is underrepresented in the VideoQA community. Recently, *Zhang et al*. [45] propose a "Rehearsal Memory" and apply it to different sequence-processing problems, including VideoQA. Inspired by this work, we propose the EMQA task as a subtask of VideoQA, which demands for memory-based architectures.

## 3. Problem Definition

In open-ended VideoQA, a model $\mathcal{M}$ is provided with a video $v$ along with a natural language question $q$, and is supposed to output a correct answer $a$. More precisely, with $V$ being a vocabulary of (sub-)words, and $F = [0,1]^{W \times H}$ being the set of all possible images with width $W$ and height $H$, we have $v \in F^N$, $q \in V^Q$, $a = \mathcal{M}(v,q) \in V^A$, where $N$ is the number of frames in the video and $Q, A$ are the number of tokens in the question and answer, respectively. Note that during inference, $A$ is determined by $\mathcal{M}$ itself by stopping upon generation of a special "End of Sentence" token.

To precisely distinguish the EMQA task as a subtask of VideoQA, we now add further constraints, similar to the "Rehearsal Memory" formalism of [45]. First, the model is decomposed into an episodic memory formation module $\mathcal{E}$ and a question answering module $\mathcal{Q}$ as $\mathcal{M}(v,q) = \mathcal{Q}(e,q)$, where $e = \mathcal{E}(v)$ is the EM. Second, we impose a constant size constraint on the EM, *i.e*., $|e| \in O(1)$ (and crucially, $|e| \notin O(N)$), where $|e|$ represents the number of elements of representation $e$, or – more generally speaking – its storage requirement. This formulation automatically implies that the video content $v$ cannot be analyzed or filtered with respect to the question directly, since $\mathcal{Q}$ can only access the EM $e$. Conversely, the EM formation module is required to condense all potentially relevant information into the fixed-size representation $e$, thus, also leading to computation requirements of $\mathcal{Q}$ being independent of $N$.

While we do not specify a fixed number for the upper bound of $|e|$ as part of the proposed EMQA task, we note that for a fair comparison of models, it is crucial to report details on $|e|$ along with the results. Obviously, it is easier to achieve good performance as $|e|$ grows, approaching unconstrained VideoQA when $|e|$ gets bigger than the storage required for the input videos $v$. As a quantitative measure of $|e|$, we suggest to report the number of 4-byte floating point entries of $e$ in case of a latent vector representation (*i.e*., the dimensionality of $e$, as done in Sec. 6), or alternatively the storage constraint in bytes.

To further illustrate the difference between VideoQA and EMQA, we can compare it to a human watching a video. VideoQA corresponds to the setting where one gets the question before watching the video, and thus can directly analyze the video with respect to the question. In contrast, EMQA is the task of answering a question which is only revealed *after* seeing the video. As a brute-force approach would be to store the complete video (or some other linearly growing representation, *e.g*., feature representations for each second), our EMQA formulation prohibits this by constraining the memory created during watching the video to a constant size, independent of the input length.

## 4. Dataset

We introduce the QAEGO4D dataset by explaining where we obtained videos and questions, how we collected natural language answers and finally presenting some statistics.

**Videos and Questions** We build the QAEGO4D dataset as an extension of Ego4D [9]. Specifically, each sample $s = (v, q, a, \tau)$ in our new dataset consists of a video $v$, natural language question $q$ and answer $a$, as well as an annotation $\tau = (\tau_{start}, \tau_{end})$ of a temporal window in which the answer can be deduced from the video. As one of the challenges associated with the Ego4D dataset, *Grauman et al*. [9] propose the EM – NLQ task, where $v$ and $q$ are given and $\tau$ should be inferred. Thus, we only need to additionally collect $a$ for constructing QAEGO4D. We filter out samples with "When?" questions, as there would be no well-defined natural language answer for them. Furthermore, Ego4D provides dense text narrations for each video, *i.e*., each narration $n$ is defined by a text $t$ as well as temporal window

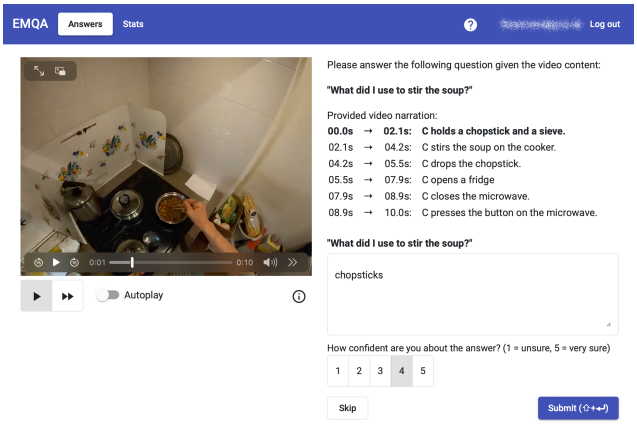


Figure 2. A screenshot of the website used to collect the QAEGO4D dataset.

$\theta = (\theta_{start}, \theta_{end})$. We utilize this to ease data collection, as described below.

**Answer Annotation** We employed human annotators to give possible answers to each question from the Ego4D EM – NLQ task. Since the ground truth annotation of the relevant temporal frame $\tau$ is already provided by the NLQ data, we utilize this and show the annotators only the corresponding snippet $\tau'$ of each video, where $\tau' = (\tau_{start} - 5s, \tau_{end} + 5s)$ adds ten seconds of context so that it is easier for human subjects to understand the video. In contrast to the length of videos $v$ which is 8.3 minutes on average, the answer snippet $v[\tau']$ is only 19.5 seconds on average, thus, significantly reducing the data collection overhead. Additionally, annotators see the text narrations $n$ which overlap with the answer snippet, *i.e.*, where $\tau' \cap \theta \neq \varnothing$. This is to help answering the question and ensuring consistent word choices when formulating the answer. For "Who?" questions, we ask annotators to answer with a short description of the referenced person, *e.g.*, "the woman in black", instead of reusing the identifiers ("person X") from the text narrations. Before submitting, annotators had to choose how confident they are with their answer. Unanswerable questions could be skipped. A screenshot of the annotation tool can be seen in Fig. 2.

**Data Analysis** After collecting the answer annotations, we split up the dataset into train, validation and test. The train set aligns with the train split of the Ego4D NLQ annotations. Since the Ego4D test data is not published, we split up the canonical videos in the validation set, and use half of them for testing. Tab. 2 shows the sizes of each split of QAEGO4D.

On exporting the data from the data collection site, answers were normalized by converting to lower case, stripping whitespaces and removing trailing punctuation. 166 samples were excluded because they were skipped by three annotators, and 404 further samples were removed from the dataset by simple filtering rules based on the question type and the first words of the answer (e.g. "Who..." with answer "on the ..."). Additionally, a dictionary-based spell checker[2] was used to detect typos and spelling mistakes, which were then manually corrected by the authors (351 samples in total). In total, QAEGO4D contains 4837 unique answers, of which 3740 occur in the training set. These unique training answers account for only 64.4% and 67.0% of answers in the validation and test split, respectively, indicating the importance of the open-ended, generative formulation of VideoQA. The most frequent answers can be seen in Fig. 3. Furthermore, Fig. 4 shows the distribution of answer lengths across all data splits. The spike for answer length three can be explained by the common "in/on/at the X" answers to the "Where...?" questions.

| | train | valid | test |
|---|---|---|---|
| #videos | 997 | 162 | 166 |
| #QA pairs | 10746 | 1913 | 1854 |

Table 2. QAEGO4D split sizes.

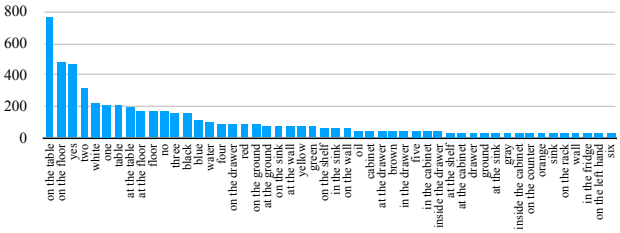


Figure 3. Histogram of 50 most frequent answers accross all splits of the QAEGO4D dataset.

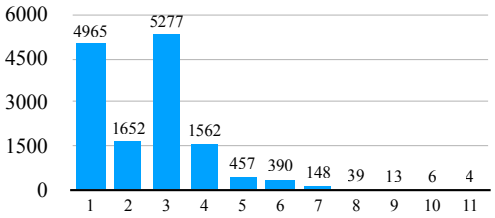


Figure 4. Histogram of answer lengths (number of words) in QAEGO4D.

## 5. Methods

In this section, we first present the baseline models we apply to the unconstrained VideoQA task, and then

[2] https://github.com/barrust/pyspellchecker

proceed with the memory-based models evaluated on the EMQA task. All models have in common that they operate on a temporal sequence of feature vectors instead of the raw video data. If not noted otherwise, we use the pre-extracted Slowfast 8x8 ResNet101 [7, 11] features provided by Ego4D [9], which encode the video with window size and stride of 32 and 16 frames, respectively.

### 5.1. VideoQA

**JustAsk** [38] is a recent model reaching state-of-the-art results on existing VideoQA benchmarks by utilizing extensive pretraining on automatically generated VideoQA data. We use the pretrained iVQA checkpoints provided by the authors and fine-tune the network on QAEGO4D. Video feature extraction and model training is done using the code[3] provided by the authors of JustAsk. Importantly, this model does not perform open-ended, generative question answering. Instead, it chooses from a fixed answer vocabulary, for which we use the set of all answers present in the QAEGO4D train set.

**HCRN** [19] is another recent VideoQA model. Using the codebase[4] of the authors, we train this model from scratch on QAEGO4D , extracting appearance and motion features from the data as defined in their code. Similar to JustAsk, HCRN uses a fixed answer vocabulary, which is constructed the same way as described above.

**SimpleVQA** is a very simple baseline model based on the Transformer [35] encoder-decoder architecture, specifically using a pretrained T5 [29] model (`t5-base` checkpoint). We first transform each video feature using a linear layer to match the hidden dimension of the transformer. The question is passed to the transformer encoder and the resulting representation is concatenated with the transformed visual sequence to form the context sequence. Subsequently, the decoder generates the answer in an auto-regressive manner, using its cross-attention to attend to the context sequence.

**Longformer** [1] is a transformer model specifically designed to handle very long input sequences by constraining self-attention to a local window around each position and some privileged "global" tokens, instead of building the full-scale attention matrix which scales quadratically with the input length. We apply this model to VideoQA on QAEGO4D in the following way: First, the video features get projected to match the embedding dimension of the Longformer. Then, these features and the embedded question tokens are concatenated and passed to the Longformer, where the positions corresponding to question tokens are allowed to perform and receive "global" attention, whereas the video tokens only attend to their local window (and the global tokens). The Longformer's output sequence of hidden states is then passed to the cross-attention input of an auto-regressive T5-decoder, which produces the answer.

**BlindVQA** is the most simple baseline, which is purely a T5 encoder-decoder model. It does not receive the visual input, and has to guess the answer based on the question only. This accounts for dataset-specific bias, as answers to some questions might be easy to guess (*e.g.*, "Where is the stove?" $\rightarrow$ "kitchen").

### 5.2. EMQA

As explained in Sec. 3, building an EMQA model involves defining the episodic memory formation module $e = \mathcal{E}(v)$ and a question answering module $a = \mathcal{Q}(e, q)$. All baseline EMQA models share a common architecture for $\mathcal{Q}$, for which we again use pretrained T5 [29] networks. Specifically, the episodic memory $e$ is concatenated with the sequence of hidden states produces by the T5 encoder, and the decoder attends to the resulting sequence using its cross-attention during auto-regressive answer generation. However, we note that the EMQA setting does not constrain the specifics of $\mathcal{Q}$, as long as it only has access to the fixed-size episodic memory $e$ instead of the full-length video $v$. More advanced architectures for $\mathcal{Q}$ are left open for future work, and we focus on different modules for $\mathcal{E}$, which are introduced in the following.

**DNC** [10] and **STM** [18] can both be immediately used as $\mathcal{E}$. However, with the very long input videos, this would result in extremely slow training times. To tackle this problem, we first project down the video feature dimension from $h_f$ to $h_s$, then split up the video input sequence into fixed-size segments of length $l$, and then feed the MANN with each time step corresponding to one flattened segment, where the input size to the MANN thus is $s \cdot h_s$. For DNC, we directly use the memory matrix $M$ (in the notation of [10]) as $e$, whereas for STM, the output vector $o_T$ for the final time step $T$ is used as $e$, as using the item or relation memory would require additional transformations.

**LT-CT** is a Long-Term (LT) extension of the Compressive Transformer (CT) [28]. CT reads the input sequence split up into fixed-size segments. Each segment is processed by a transformer network, which has a recurrent connection to the previous layer's hidden states from the previous time step. Previous hidden states are kept in a FIFO queue, and the oldest items at each time step get compressed and put into another queue of compressed memories. In turn, the oldest vectors from the compressed memory are dropped. Thus, the CT itself is not able to handle infinite context

[3] https://github.com/antoyang/just-ask
[4] https://github.com/thaolmk54/hcrn-videoqa

lengths. Therefore, we add a simple on-layer recurrent unit in form of an LSTM cell to each layer of the CT, building the LT-CT. Furthermore, we enhance the architecture by using multiple compression levels, with each one receiving the vectors dropped from the previous one. After reading in the complete video sequence, the episodic memory $e$ is defined as the flattened sequence of all the vectors in the memory, all compressed memories, and the LSTM cell states.

**RM** (Rehearsal Memory) [45] is another model specifically fitting to EMQA. Similar to the above models, the video input sequence is split up into fixed-size segments. Each segment is encoded by a transformer encoder, whereupon a recurrent component based on GRUs attends to the current segment to produce the new memory state. The set of memory cells naturally form the episodic memory $e$. Importantly, in addition to the usual language modelling loss, this model uses self-supervised rehearsal training. For this purpose, it picks segments from the video input and tries to restore them from the final memory of the model. To pick meaningful segments during this process, RM utilizes attention scores of an unconstrained VideoQA model as teacher for selecting which segments are important. For this, we use the a model similar to SimpleVQA described above, with the modification that it receives input segments instead of the individual items of the sequence.

### 5.3. Utilizing NLVL Supervision

To fully exploit the QAEGO4D data, models can use the ground-truth temporal window annotations as an additional source of supervision. We add this to two of our baselines models in the following simple way:

For **SimpleVQA+** and **Longformer+**, we extract the transformer decoder's cross-attention scores for the positions belonging to the video input. This subset of scores is passed through softmax again to produce the distribution of attention on the video only. Then, we inject the NLVL supervision as a ranking loss on the attention scores as done in the STAGE [21] model. Specifically, for each input position part of the target moment (positive samples), we sample two positions from somewhere else in the video (negative samples), and then apply LSE [22] loss between the positive and negative samples. The final loss $\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda * \mathcal{L}_{\text{NLVL}}$ is the weighted sum of language modeling loss $\mathcal{L}_{\text{LM}}$ and NLVL loss $\mathcal{L}_{\text{NLVL}}$, where we empirically choose $\lambda = 10$ for SimpleVQA+ and $\lambda = 1$ for Longformer+.

## 6. Experiments

### 6.1. Metrics

Since we phrase the VideoQA problem as open-ended, generative QA, solely using the "plain" accuracy as a performance metric, *i.e.*, the percentage of answers where the model generates exactly the ground truth solution, is not sufficient. Therefore, in addition to accuracy, we report several standard machine translation metrics as done in previous work [44]. This includes BLEU-4 [26], METEOR [17], ROUGE-L (f-score) [23].

| | Acc. | BLEU | METEOR | ROUGE |
|---|---|---|---|---|
| BlindVQA | 9.0 | 3.6 | 17.4 | 25.9 |
| SimpleVQA | 9.3 | 6.1 | 17.4 | 26.1 |
| Longformer [1] | 3.0 | 2.4 | 15.4 | 20.9 |
| HCRN$^{\star}$ [19] | 10.3 | 7.6 | 17.2 | 25.7 |
| JustAsk$^{\dagger\star}$ [38] | 9.6 | 3.9 | 17.8 | 26.7 |
| SimpleVQA+ | 9.7 | 3.6 | 18.3 | 27.1 |
| Longformer+ [1] | 6.7 | 5.4 | 16.9 | 24.4 |

Table 3. VideoQA results on the QAEGO4D test set. The models in the upper part of the table only use the supervision from the answer annotations. Models in the lower part additionally have access to NLVL temporal window annotations. $^{\dagger}$pretrained on additional VideoQA data. $^{\star}$ non-generative question answering.

### 6.2. Experimental Settings

For the open-sourced JustAsk and HCRN models, we use the code and settings as provided by the authors, solely changing the learning rate for JustAsk to $10^{-4}$. For all other models, we use PyTorch Lightning[5] to train each on the train set of QAEGO4D with the Adam [15] optimizer without weight decay and automatically select the learning rate for each experiment using Auto LR Tuning [31]. All experiments use a fixed random seed and an effective batch size of 32 samples (taking multi-GPU training and gradient accumulation into account). Early stopping based on the validation language modeling loss with a patience of ten validation steps is used, and the checkpoint from the best epoch on the validation set is selected. When there are multiple configurations for one architecture, we use the one with the higher validation ROUGE score as produced by the selected checkpoint. Final results are reported on the test set.

### 6.3. Results

**VideoQA** Tab. 3 shows the results of evaluating the VideoQA models on the QAEGO4D test set. It can be seen that the general level of performance is quite low, as the dataset is extremely challenging considering the length of the videos (8 min. on avg.) and the diversity of the depicted scenarios and questions. This level of challenge is expected when looking at the NLVL baseline results of *Grauman et al*. on the Ego4D NLQ dataset we build QAEGO4D upon [9, Tab. 11]. Since JustAsk and HCRN have a restricted

[5] https://github.com/PyTorchLightning/pytorch-lightning

| | $\#e$ | Acc. | BLEU | METEOR | ROUGE |
|---|---|---|---|---|---|
| DNC [10] | 16 | 9.7 | 3.4 | 17.9 | 27.0 |
| STM [18] | 1 | 9.4 | 5.8 | 17.6 | 26.2 |
| LT-CT [28] | 377 | 10.5 | 5.3 | 18.5 | 27.5 |
| RM [45] | 16 | 9.9 | 4.5 | 17.7 | 26.6 |

Table 4. EMQA results on the QAEGO4D test set. $\#e$ measures the number of 768-dimensional vectors constituting the episodic memory vector, *i.e.* $|e| = \#e \cdot 768$.

answer vocabulary, they have a general advantage with respect to the other models which honor the more realistic generative VideoQA conditions when comparing the plain accuracy. Furthermore, both these models use architectures specialized for VideoQA. The additional advantage of being pretrained on larger (non-egocentric) VideoQA datasets does not pay off significantly for JustAsk, as it outperforms HCRN on METEOR and ROUGE, but is worse than HCRN in plain accuracy and BLEU score.

While the SimpleVQA model performs only slightly better than the guessing baseline, an improvement on all metrics except for BLEU from training with NLVL loss (SimpleVQA+) can be observed. Despite the extremely poor overall performance of Longformer, the same observation can be made when comparing with Longformer+. This indicates that utilizing the temporal window annotations in QAEGO4D indeed provides useful additional supervision to the VideoQA task, which should be further utilized in future work.

**EMQA** The results of evaluating the EMQA models can be seen in Tab. 4. For a fair EMQA comparison, however, it is crucial to look at the size of the episodic memory used in a model. Since all our models use a memory vector hidden size of 768 (to align with the hidden size of the `t5-base` decoder), we report the number of memory vectors $\#e$ instead of the number of floating point entries $|e| = \#e \cdot 768$ in Tab. 4. STM uses only one hidden vector in our simple implementation (the output of the last time step), while DNC and RM both use 16 memory cells in our experiments. Thus, the low performance of STM is reasonable. RM and DNC share the same $|e|$, however, there is no model clearly beating the other one, as both outperform each other in two of the metrics.

In contrast, LT-CT uses all its queued hidden states as well as the LSTM cell states as episodic memory, which results in a total of 377 vectors in our configuration. As EMQA is a more restrictive setting than VideoQA, the performance is generally expected lower. Nevertheless, LT-CT beats the other EMQA baselines as well as the VideoQA models in most metrics. Likewise, RM and DNC perform on a similar level as most of the VideoQA baselines. This might indicate that constraining the memory could potentially be useful for handling long inputs as it forces the model to select relevant information appropriately. Future work should further investigate this issue.

Despite these observations, the overall performance on both VideoQA and EMQA is still very low and close to guessing, as the distance to the blind baseline is rather small even for the top-performing models. Future work will need to come up with more effective strategies for analyzing very long videos as well as representing a video with constant size constraints without knowing the question in advance.

## 7. Conclusion

We present the QAEGO4D dataset, featuring egocentric videos annotated with questions, answers and relevant temporal windows. Moreover, we propose to use this data not only for VideoQA, but also for EMQA, where a model is constrained to keep only a constant amount of memory of an arbitrarily long input video, and then reason solely based on this memory afterwards. Finally, we present various memory-augmented baseline models to tackle the EMQA task, and compare them by presenting extensive experiments on the new dataset. The results both on unconstrained VideoQA as well as EMQA highlight the extremely challenging nature of QAEGO4D and provide a baseline to improve upon by future work.

## Acknowledgments

This work has been supported by the German Federal Ministry of Education and Research (BMBF) under the project OML (01IS18040A). We thank Datoid, LLC and all annotators for enabling the fast data collection.

## References

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv*, 2020. 2, 5, 6

[2] Leonard Bärmann, Fabian Peller-Konrad, Stefan Constantin, Tamim Asfour, and Alex Waibel. Deep episodic memory for verbalization of robot experience. *IEEE RA-L*, 6(3):5808–5815, 2021. 1, 2

[3] Santiago Castro, Mahmoud Azab, Jonathan Stroud, Cristina Noujaim, Ruoyao Wang, Jia Deng, and Rada Mihalcea. LifeQA: A real-life dataset for video question answering. In *LREC*, pages 4352–4358, 2020. 2

[4] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *PMLR*, volume 139, pages 1931–1942, 2021. 2

[5] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS Worksh. Deep Learn.*, 2014. 3

[6] Chenyou Fan. EgoVQA - an egocentric video question answering benchmark dataset. In *ICCVW*, pages 4359–4366, 2019. 2

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 5

[8] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In *ICCV*, pages 5277–5285, 2017. 3

[9] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4D: Around the world in 3,000 hours of egocentric video. *arXiv*, 2021. 1, 2, 3, 5, 6

[10] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016. 1, 2, 3, 5, 7

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 3

[13] Hartwig Holzapfel, Thomas Schaaf, Hazim Kemal Ekenel, Christoph Schaa, and Alex Waibel. A robot learns to know people - first contacts of a robot. In *Proc. Ann. German Conf. on Artif. Intel.*, 2006. 1

[14] Junyeong Kim, Minuk Ma, Trung Pham, Kyungsu Kim, and Chang D. Yoo. Modality shifting attention network for multimodal video question answering. In *CVPR*, 2020. 1

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[16] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, pages 706–715, 2017. 3

[17] Alon Lavie and Abhaya Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. ACL, 2007. 6

[18] Hung Le, Truyen Tran, and Svetha Venkatesh. Self-attentive associative memory. In *PMLR*, volume 119, pages 5682–5691, 2020. 2, 3, 5, 7

[19] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for multimodal video question answering. *IJCV*, 129(11):3027–3050, 2021. 2, 5, 6

[20] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. TVQA: Localized, compositional video question answering. In *EMNLP*, pages 1369–1379, 2018. 2

[21] Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. TVQA+: Spatio-temporal grounding for video question answering. In *ACL*, pages 8211–8225, 2020. 6

[22] Yuncheng Li, Yale Song, and Jiebo Luo. Improving pairwise ranking for multi-label image classification. In *CVPR*, 2017. 6

[23] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. ACL, 2004. 6

[24] Minuk Ma, Sunjae Yoon, Junyeong Kim, Youngjoon Lee, Sunghun Kang, and Chang D. Yoo. VLANet: Video-language alignment network for weakly-supervised video moment retrieval. In *ECCV*, pages 156–171, 2020. 3

[25] Taiki Miyanishi and Motoaki Kawanabe. Watch, listen, and answer: Open-ended VideoQA with modulated multi-stream 3d ConvNets. In *EUSIPCO*, pages 706–710, 2021. 2

[26] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318, 2002. 6

[27] Devshree Patel, Ratnam Parikh, and Yesha Shastri. Recent advances in video question answering: A review of datasets and methods. In *ICPR Worksh.*, pages 339–356, 2021. 2

[28] Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *ICLR*, 2020. 2, 5, 7

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020. 5

[30] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics*, 1:25–36, 2013. 3

[31] Leslie N. Smith. Cyclical learning rates for training neural networks. In *WACV*, pages 464–472, 2017. 6

[32] Guanglu Sun, Lili Liang, Tianlin Li, Bo Yu, Meng Wu, and Bolun Zhang. Video question answering: a survey of models and datasets. *Mob. Netw. App.*, 26(5):1904–1937, 2021. 2

[33] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding stories in movies through question-answering. In *CVPR*, pages 4631–4640, 2016. 2

[34] Endel Tulving. Episodic and semantic memory. *Organization of memory*, 1:381–403, 1972. 1

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 6000–6010, 2017. 2, 5

[36] Alex Waibel, Hartwig Steusloff, Rainer Stiefelhagen, and Kym Watson. *Computers in the human interaction loop*. Springer, 2009. 1

[37] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM MM*, pages 1645–1653, 2017. 1, 2

[38] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*, pages 1686–1697, 2021. 1, 2, 5, 6

[39] Weijiang Yu, Haoteng Zheng, Mengfei Li, Lei Ji, Lijun Wu, Nong Xiao, and Nan Duan. Learning from inside: Self-driven siamese sampling and reasoning for video question answering. In *NeurIPS*, page 13, 2021. 1, 2

[40] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. ActivityNet-QA: A dataset for understanding complex web videos via question answering. *AAAI*, 33(1):9127–9134, 2019. 1, 2

[41] Heeseung Yun, Youngjae Yu, Wonsuk Yang, Kangil Lee, and Gunhee Kim. Pano-AVQA: Grounded audio-visual question answering on 360° videos. In *ICCV*, pages 2011–2021, 2021. 2

[42] Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. Natural language video localization: A revisit in span-based question answering framework. *IEEE TPAMI*, pages 1–1, 2021. 3

[43] Songyang Zhang, Houwen Peng, Jianlong Fu, Yijuan Lu, and Jiebo Luo. Multi-scale 2d temporal adjacent networks for moment localization with natural language. *IEEE TPAMI*, 2021. 3

[44] Yizhe Zhang, Siqi Sun, Xiang Gao, Yuwei Fang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. Joint retrieval and generation training for grounded text generation. In *AAAI*, 2022. 6

[45] Zhu Zhang, Chang Zhou, Jianxin Ma, Zhijie Lin, Jingren Zhou, Hongxia Yang, and Zhou Zhao. Learning to rehearse in long sequence memorization. In *PMLR*, volume 139, pages 12663–12673, 2021. 1, 2, 3, 6, 7