Autonomous Robots (2023) 47:1103–1118
https://doi.org/10.1007/s10514-023-10134-4

Learning to summarize and answer questions about a virtual robot's past actions

Chad DeChant[1] · Iretiayo Akinola[2] · Daniel Bauer[1]

Received: 1 May 2023 / Accepted: 3 August 2023 / Published online: 16 November 2023

Abstract
When robots perform long action sequences, users will want to easily and reliably find out what they have done. We therefore demonstrate the task of learning to summarize and answer questions about a robot agent's past actions using natural language alone. A single system with a large language model at its core is trained to both summarize and answer questions about action sequences given ego-centric video frames of a virtual robot and a question prompt. To enable training of question answering, we develop a method to automatically generate English-language questions and answers about objects, actions, and the temporal order in which actions occurred during episodes of robot action in the virtual environment. Training one model to both summarize and answer questions enables zero-shot transfer of representations of objects learned through question answering to improved action summarization.

Keywords Summarization · Interpretability · Representation learning · Long horizon tasks

1 Introduction

Autonomous robots will soon be deployed in large numbers performing a wide variety of tasks. They will operate for long periods of time, often far from their users, making real time supervision of their activities impractical. They will be faced with challenging situations and have to make decisions and perform actions on their own, all without the aid or immediate knowledge of their operators. Robot autonomy, then, presents an important challenge: the need to inform robots' operators what they have done.

Upon returning home after tasking a robot with cleaning the house, making dinner, and taking care of the dog, a robot user would like to know what happened during the day: how the dog fared, what parts of a house could not be cleaned and why, and what the robot made for dinner, for example. A farmer using robots to harvest crops would want to be able to get ask how much has been harvested, what the conditions in the field were, and if any evidence of disease and drought was seen.

It might be thought that a robot could simply keep a log of all it has done and thus give a full report of every turn, move, and decision it made during its operation. However, there are two problems with such a scenario. First, such reporting may not be possible, particularly if an agent's actions are not the result of interpretable internal planning using discrete primitives but are instead the result of following a reinforcement learning policy implemented as a neural network which simply outputs, for example, rotations and joint movements to a robot's wheels, arms, etc. Second, even if such a complete record of action existed and was human readable, it would not be useful; it would be far too long and detailed to read and make sense of in a reasonable time in any realistic situation. Instead, it will be necessary for agents to summarize their activities. And in order to make such a summary available in a format for humans to comprehend quickly and accurately it would be ideal if the summary were given in natural language.

Summaries, rather than complete records, will be particularly useful as action sequences become longer. They will also be challenging to produce because it will be necessary to identify the most important actions and, very often, to

✉ Chad DeChant
chad.dechant@columbia.edu

Iretiayo Akinola
iakinola@nvidia.com

Daniel Bauer
bauer@cs.columbia.edu

[1] Computer Science Department, Columbia University, 500 West 120 Street, New York 10027, NY, USA

[2] NVIDIA, 4545 Roosevelt Way NE, Seattle 98105, WA, USA

describe those actions using higher level abstract terms. Summaries may not fully address everything that a user wants to know about a robot's actions so a user may want to ask questions about what a robot did or saw during a particular action sequence.

Roboticists have long recognized the usefulness of being able to give natural language instructions to robots. Summarizing and answering questions about past robotic actions can be seen as a complement to instruction following. A user who gave such instructions might naturally be expected to want a short natural language summary of what was done in response to those instructions. Yet despite the volume of work that has been done on instruction following, its complement has gone largely unaddressed. Fortunately, existing and future datasets designed for instruction following tasks can be repurposed and augmented to serve as a training ground for robot action summarization and question answering. We make use of and augment the popular ALFRED dataset (Shridhar et al., 2020) which provides ego-centric video frames of episodes of robot action sequences in a virtual environment along with multiple levels of description in natural and structured language. Using a model that incorporates a large language model (LLM), we present the first work directly addressing, performing, and evaluating robot action summarization and question answering.

Our main contributions are:

Summarization of actions. We demonstrate summarization of robotic actions in both short and long summaries from video frames in a multimodal model that incorporates vision and fine-tunes a pretrained T5 LLM (Raffel et al., 2020).

Answering questions about actions. The same model is jointly trained to answer questions about robotic actions, including questions about actions performed, objects seen, and the order in which actions were performed.

Zero-shot transfer from question answering to summarization. We show that an LLM-based system trained to answer questions about held-out objects can faithfully produce summaries about those objects in a zero-shot manner, even though the objects are not in the summarization task training set. This demonstrates the transfer of representational knowledge from the question answering tasks to the summarization tasks. We further demonstrate that this transfer occurs for some question types but not others.

Automatic generation of questions and answers. We develop a method to automatically generate questions and answers using an existing dataset and its associated virtual environment and release a dataset of such questions and answers.

2 Method

Our objective is to generate a summary or question response in natural language $r \in \mathcal{L}$ of a long horizon robotic task, given the history of observations $o \in \mathcal{O}$ that the robot experienced during the task and a question or summarization prompt q. We define the robot experience/trajectory as $\tau = \{(o_0, ...)\}$. We seek to learn a function $\mathcal{F}_\theta$ such that: $r = \mathcal{F}_\theta(\tau, q)$.

2.1 Data requirements

The general problem of robot action summarization and question answering could be addressed in a variety of ways depending on the data available, the environment the robot operates in, and details of how the robot operates. A few types of data would be most helpful in training and operating an autonomous, mobile, general purpose robot to summarize its past:

(1) Ego-centric video of the robot performing tasks serves as the primary input to summarization and question answering. It can be captured by many robots and would facilitate the transfer of knowledge to new circumstances and environments.

(2) Natural and/or structured language summaries of the actions performed in the video. These summaries could be of varying lengths, depending on the needs of the end user. The presence of both short and long summaries would provide the most flexibility and choice for a user.

(3) Ground truth information about the objects and places in the ego-centric video for training question answering tasks. The objects and places present in the training dataset will determine what kinds of questions a user could subsequently ask.

2.2 Repurposed dataset

Our approach requires egocentric video or video frames, a description of an agent's actions during an episode, and information about the environment the agent operates in, particularly the locations of objects it encounters. For the purposes of the current investigation we use episodes from the ALFRED dataset. An episode of robot state-action trajectory in the original dataset has four different kinds of representation which we make use of, either as-is or transforming them in some ways. The following list of dataset elements lays out the way they are used in this work as well as noting their original purpose and description in the ALFRED dataset:

(1) *Short summaries*: Human-generated natural language one sentence summaries of the whole action sequence (called "goal descriptions" in the original dataset).

(2) *Long summaries*: High level narratives of the robotic agent's actions, provided in the original dataset in the form of action plans in the structured Planning Domain Description Language (PDDL) (McDermott et al., 1998). We convert the terms used in PDDL to natural language: for example, "GotoLocation" becomes simply "go to" and some object names become two English words instead of one word (e.g.

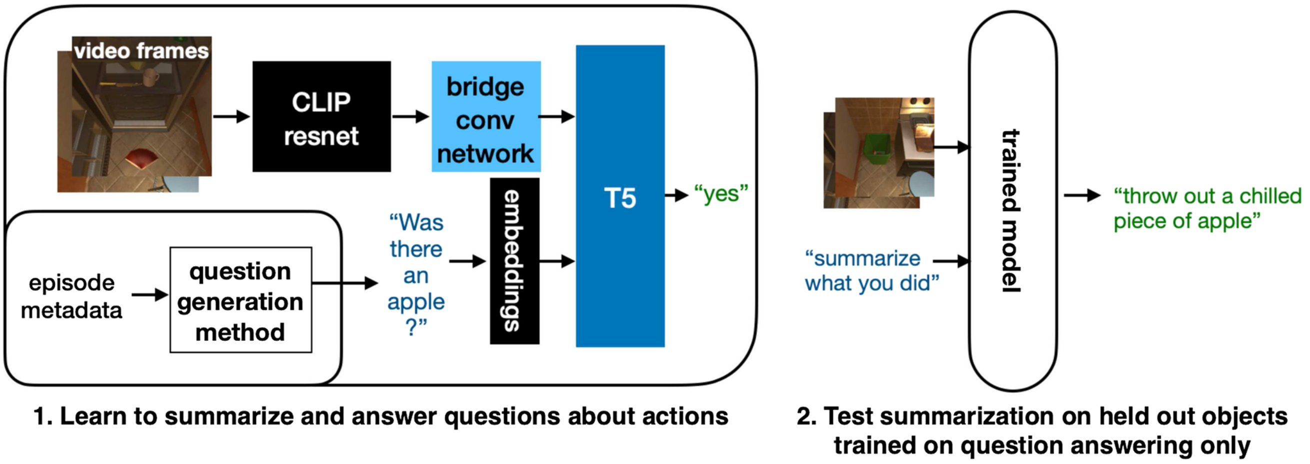

video frames
CLIP resnet
bridge conv network
T5
"yes"
episode metadata
question generation method
"Was there an apple ?"
embeddings
1. Learn to summarize and answer questions about actions
"summarize what you did"
trained model
"throw out a chilled piece of apple"
2. Test summarization on held out objects trained on question answering only

Fig. 1 Visual presentation of model and method for producing zero-shot summaries involving novel objects. Step 1 illustrates the full model: input (at the left) includes video frames as well as episode metadata describing the environment as the agent saw it. The components of the model in black (CLIP Resnet and the word embeddings) are pretrained and remain frozen during our training process, while the light blue module (the vision-to-T5 bridge network) is trained from scratch. The dark blue module, a pretrained T5, which outputs the final question answer or summary, is fine-tuned during training. Step 2 demonstrates zero-shot summarization using a previously trained model which was not trained to summarize episodes with some of the objects in the newly presented episode

"coffeemachine" becomes "coffee machine"). We also break these long summaries up to form questions, as described in the next subsection.

(3) *Natural language action description sentences*: Natural language step by step descriptions of the actions taken in each episode, written by humans, which were used as instructions in the original dataset. These are used here to form some of the questions, as described in the next section. We do not use these to generate summaries because they are too detailed and contain somewhat idiosyncratic descriptions provided by human annotators. These characteristics, which make these inappropriate to serve as ground truth summaries, nevertheless make them good training examples for naturalistic human-generated questions, which is why we use them to form the basis of questions.

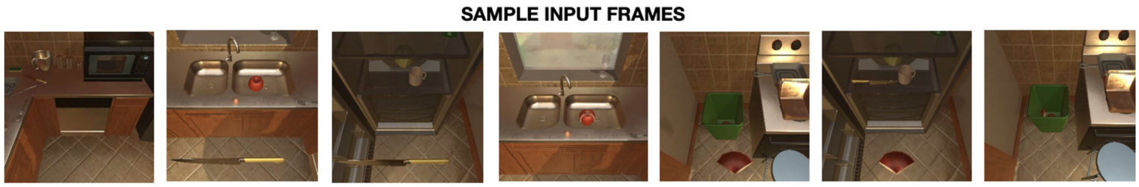
(4) *Video, images, and visual features*: Raw video of a task episode as well as a selection of still frames from the video chosen by the creators of the ALFRED datasset in such a way as to guarantee at least one still frame per low level action as defined in the original dataset. We use the pre-selected subset of still frames in the dataset, leaving the question of frame selection to future work.

Robot actions in the ALFRED dataset consist of discrete navigation and manipulation actions labeled 'low level' actions see Shridhar et al. (2020) for details; episodes have an average of 50 such actions. Because summarization and question answering involve higher level semantic descriptions, action descriptions in this work derive from two sources: PDDL converted to natural language and human annotator descriptions which include actions. The former are a restricted set (go to, pick up, put, cool, heat, clean, toggle) while the latter are unrestricted and express actions in diverse ways.

2.3 Automatic generation of questions and answers

We develop a Q&A generation algorithm that produces questions and answers about episodes of robots interacting with an environment. After initial pre-processing, the algorithm can be used in a partly online fashion during training or as a one-time off-line dataset generation step which produces a set of static questions and answers. We train models in an online fashion and provide performance metrics from the static validation sets of questions and answers we release with this work.

In addition to the elements already present in the original dataset enumerated in the previous subsection, we use the AI2THOR environment (Kolve et al., 2017) to rerun the agent trajectories for each episode in the dataset and capture information present while the agent is in the environment. At each time step after executing an action, the environment returns a 'metadata' python dictionary with information about the last action taken, the agent's current position and pose, and the objects present in the environment. Information about objects includes whether they are visible and within a specified distance of the agent (we use the default 1.5 ms). We use these two pieces of information to construct questions about whether objects were present in the environment. Though here we use one particular existing dataset and environment, our approach is general and can be applied to other datasets and environments with action descriptions in natural or structured language and available information about the environment.



Question / Prompt	Answer
OBJECT QUESTIONS	
Was there a mug?	yes
Was there an egg or an apple?	apple
SIMPLE ACTION QUESTIONS	
Did you go to the desk?	no
Did you slice the apple or wash the apple?	slice the apple
COMPLEX ACTION QUESTIONS	
Did you turn right, move to the counter in front of the dishwasher?	yes
SIMPLE TEMPORAL QUESTIONS	
What did you do just before go to the garbage can?	cool the apple
What did you do just after put the knife in the fridge?	go to the sink basin
COMPLEX TEMPORAL QUESTIONS	
What did you do just before turn right, move to the bin that is left of the stove?	cool the apple
What did you do just after place the knife inside the refrigerator on the left of the mug?	go to the sink basin
LONG SUMMARIES	
Narrate what you did.	go to the countertop, pick up the knife, go to the apple, slice the apple, go to the fridge, put the knife in the fridge, go to the sink basin, pick up the apple, go to the fridge, cool the apple, go to the garbage can, put the apple in the garbage can
SHORT SUMMARIES	
Summarize what you did.	throw out a chilled piece of apple

Fig. 2 Sample partial selection of input frames from an episode in a seen environment originally from the ALFRED dataset (at the top), generated questions (on the left, in blue) and expected answers (on the right, in green), broken up into question type, along with the prompts for long and short summaries, at the bottom

The algorithm produces nine types of questions in three broad categories (see Fig. 2 for examples of each type from the valid seen set and Appendix F for additional examples from the valid unseen set):

(1) **Object questions** about the presence of objects in the environment, both those the agent interacted with and those it only saw. There are two kinds of object question: "object yes/no" questions of the form, "was there an <object>?", which require only "yes" or "no" answers and "object either/or" questions of the form, "was there an <object A> or <object B>?" which require the model to output the name of the object present. Our algorithm uses the metadata of all objects visible in the environment to ensure that only one of the objects in an either/or question will have been seen during an episode. The algorithm samples objects with negative answers in proportion to their appearance in the dataset so that the model cannot, for example, learn to always answer, "no", for seldom-seen objects. Questions with "yes" and "no" answers are presented with equal frequency.

(2) **Action questions**, which ask about actions the agent performed. The two types of question—"action yes/no" and "action either/or"—follow the structure of the respective object questions explained above. There are two subtypes of the "action yes/no" questions: "simple action yes/no" uses the relatively simple language converted from PDDL for both the questions and answers. "Complex action yes/no" uses the raw human-generated description of each action step to pose the "yes/no" question. "Action either/or" questions present an either/or choice between two actions described in the simpler language of the converted PDDL plans.

(3) **Temporal questions** about the order in which actions were performed, of two primary kinds. The first kind—"just before" questions—asks what action was performed immediately before a named action ("what did you do just before <action description>?") while the second—"just after" questions—asks what action was performed immediately following the named action ("what did you do just after <action description>?"). If an action occurs more than once in an episode it will not appear in a temporal question to avoid ambiguity.

Each of these types of temporal questions has two subtypes. The first is asked using the simpler description of actions from converted PDDL while the second uses a human-generated action description sentence to formulate the question. Human-generated descriptions are longer, contain more diverse word choice, and sometimes mention irrelevant details. The answers to both question subtypes are in the simpler action description format. We suggest that this distinction between enabling the model to answer both simple and more complexly-worded questions while only answering in simpler language is desirable because while a robot agent should be able to understand questions phrased in a variety of ways, such an agent should not produce similarly varied answers, but instead generate only simple, consistent language.

In addition to these questions and answers, we also prompt the model to produce two kinds of summaries:
(1) **Short summaries** are the short one sentence descriptions of the action sequences written by human annotators as provided in the original dataset. We train the model to output a summary of a given episode with the text prompt, "summarize what you did."
(2) **Long summaries**, which are the longer narratives of actions converted from PDDL to natural English. Although these are meaningfully longer than the one sentence summaries, they are significantly shorter than a step by step account of every low level action the virtual robot performed (e.g. move ahead, turn, look up, etc.). The model is trained to output a long summary of an episode with the prompt, "narrate what you did."

2.3.1 Dataset of questions and answers

We will release both the code to generate the questions and answers as well as a static set of premade questions and answers aligned to episodes in the ALFRED dataset. The static dataset was generated to produce up to ten question instances per question type for each episode; in some cases there are fewer than ten such question instances per episode because not all question types can produce ten question instances for a given episode.

The entire static question and answer set contains 486,704 questions paired to episodes in the ALFRED dataset's training set, 18,891 questions paired to its seen environments validation set, and 19,097 in its unseen environments validation set.

2.4 Joint summarization and question answering model

We present a learned algorithm that takes as input ego-centric video frames of a virtual mobile robot along with a natural language question or summarization prompt and produces an answer or summary in response.

Our full neural network model (see the breakdown on the left in Fig. 1) combines several components. Video frames from each episode are fed as individual images collected into a batch into a frozen Resnet network (He et al., 2016) pretrained as part of the CLIP model (Radford et al., 2021). We extract the output of the last convolutional layer and feed it into a three layer convolutional network trained from scratch, which acts a bridge network between the Resnet and the next step in the pipeline, a pretrained T5 transformer LLM (Raffel et al., 2020) ("t5-base"in the Hugging Face library (Wolf et al., 2020)). The bridge network outputs one vector for each input image; these vectors are concatenated together along with the tokenized question or summary prompt which is embedded using the T5 model's pretrained embeddings as the input to the T5 model. The bridge network serves to translate the input from the CLIP latent space into one which can be processed by the T5. We find that fine tuning the entire T5 – rather than leaving either or both of the encoder or decoder frozen – leads to better results. While the T5 model was pretrained only on language data, we use it for simultaneous language and visual input, following other work which has shown the ability of language model transformers to process multimodal data (Lu et al., 2022; Tsimpoukelli et al., 2021). The latent space of inputs which the T5 expects is likely also modified during this fine tuning, so the adaptation of the T5 to process multimodal input can be seen as a result of both the bridge network and the fine tuning process.

As the T5 is an encoder-decoder model it is able to generate encoded representations of the images conditioned on the given question or prompt. We train a single model to answer all questions and produce long and short summaries so that it must learn to generate representations useful for all of these tasks. During an epoch of training we iterate through each episode in random order. For each episode, the model must produce long and short summaries and answer one question of each of the nine question types (when such a question exists for that episode).

2.5 Zero-shot summarization after question answering

We are interested in the possible interaction between question answering and summarization abilities within the model, in particular if representations of objects transfer between these tasks. We therefore alter the training regime to leave some objects out of the summarization training set and measure whether the model is still able to produce accurate summaries about interactions with the objects. In these experiments, we first randomly select a set of five objects from among the most common thirty objects in the dataset (excluding the top ten). We then identify all episodes whose long summaries contain those objects (i.e. any episode in which the virtual robot interacts with those objects) and set them aside as a 'held-out' set. The model is then trained on questions and answers involving all episodes, including the held-out episodes, but is not trained to produce either long or short summaries of

Table 1 Accuracy and precision scores for question and summary outputs by output type, including standard deviation.

Question / prompt	Seen envs accuracy	Precision	Unseen envs accuracy	Precision
Object yes/no	.954 ±.007	–	.907 ±.010	–
Object either/or	.990 ±.003	.990 ±.003	.966 ±.009	.968 ±.010
Simple action yes/no	.975 ±.001	–	.892 ±.004	–
Complex action yes/no	.935 ±.003	–	.895 ±.004	–
Simple action either/or	.988 ±.003	.995 ±.001	.923 ±.019	.963 ±.009
Simple action just before	.948 ±.004	976 ±.003	.865 ±.012	.957 ±.004
Complex action just before	.927 ±.007	.967 ±.004	.818 ±.013	.939 ±.005
Simple action just after	.959 ±.002	.983 ±.002	.815 ±.017	.936 ±.004
Complex action just after	.911 ±.009	.952 ±.004	.730 ±.015	.887 ±.002
Long summary	.850 ±.005	.969 ±.011	.475 ±.035	.945 ±.004
	ROUGE	BLEU	ROUGE	BLEU
Short summary	.571 ±.000	.556 ±.006	.517 ±.004	.504 ±.022
Long summary	.981 ±.000	.969 ±.000	.922 ±.004	.880 ±.010

ROUGE and BLEU scores also given for summaries. Results shown are from two validation sets: those based on episodes in virtual environments seen during training are on the left, unseen environments on the right. None of the actual episodes themselves, of either type, are found in the training set. Precision scores are not shown for "yes/no"answers where such scores must equal the accuracy scores. Results are averaged from three models with different random seeds, all tested on the set of static held-out questions

the held-out episodes. We then test its ability to summarize these held out episodes.

3 Results

3.1 Summarization and question answering

We find that our model performs very well on both short and long summarization tasks and on the questions from our Q&A generation algorithm. Table 1 presents results for all question and summarization types. An answer is considered accurate if it completely matches the target answer. BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) scores are also given for the two summary types. The BLEU score is a measure of how well the generated text matches the ground truth text, penalizing words and phrases which are not present in the ground truth while ROUGE measures how much of the ground truth text is present in the generated text, penalizing words and phrases which are missing from the generated text. Unigram precision scores measure the percentage of generated words which are in the ground truth text and are given for question answering tasks which require more than one word as an answer. As the short summaries are more lexically diverse, binary accuracy measures are less appropriate so only BLEU and ROUGE scores are given for the short summaries.

A few patterns in the results can be seen. First, the performance generally varies depending on how much generated text must be produced in an answer. Longer answers provide more opportunities for errors so performance when measured by the strict metric of complete accuracy tends to be worse. This is particularly true for the question which asks for a long summary of the agent's action, which has the worst results according to the all-or-nothing accuracy metric.

Table 2 Overlap of missing objects between questions and long summaries by question type, averaged over three models tested on the static held out valid unseen set.

Question	Error overlap
Object yes/no	.054 ±.023
Object either/or	.061 ±.029
Simple action yes/no	.370 ±.059
Complex action yes/no	.222 ±.051
Simple action either/or	.464 ±.058
Simple action just before	.441 ±.071
Complex action just before	.414 ±.028
Simple action just after	.725 ±.020
Complex action just after	.414 ±.028

Overlap here is the number of missing word errors per question type for which the long summaries are also missing the same word in the same episode, as a percentage of all missing word errors per question type

Second, "either/or" questions have better accuracy than their corresponding "yes/no" questions. This could be because asking if, for example, an action was performed is made easier when it is a choice between two actions so that any uncertainty the model has about one of the actions may be offset by its certainty about the other option. It is also possible that the model has a harder time connecting the meaning of the "yes/no" answers back to the input, particularly since most of the questions require outputting an object or action name, not just a “yes/no”.

Third, it might be expected that questions about the order that actions took place would be significantly more difficult for the model to interpret than those about the mere occurrence of those actions. Surprisingly, then, we find that in most cases the model’s performance on temporal questions is very similar to that on the other questions.

The model tends to make two kinds of errors when generating anything other than “yes/no” answers. It sometimes misidentifies objects, especially small ones, and particularly in the unseen environments. It also sometimes uses a different description for a location than the ground truth annotation, in some cases doing so in a way that is nevertheless consistent with the action as seen in the episode. For example, the ground truth annotation may read, “go to the apple” while the model outputs, “go to the counter” when the apple is on the counter. See Fig. 3 for examples of errors in short and long summaries generated by the model.

The errors made by the model display some consistency between the different questions asked and between the questions and summaries. For example, in one episode of the validation seen set which involves moving a book, it consistently mistakes the book for a pen, answering a “just before” question with, “put the pen on the desk," producing a short summary, “put two pens on the right side of the desk,” and beginning the long summary with, “go to the side table, pick up the pen...” There is a marked difference in the consistency of these errors depending on question type, however, as we show in Table 2. We measure this consistency by counting what fraction of particular objects omitted from the model’s answers to a given question type is also missing from the corresponding long summaries about that episode. This fraction is compared for different question types. We find that questions which require generating both an action and an object together have the highest degree of overlap in which objects they fail to identify and which are also missing in the long summaries; the temporal “just before / just after” answers in particular show high consistency with the long summaries. We hypothesize that the representations which the model uses for summarization align better with those it uses for the question types where there is higher overlap of missing words.

3.2 Zero-shot summarization via question answering

Can question answering improve the ability to summarize? We find that when the model is trained to answer questions about episodes involving all objects, it is then able to go on to summarize episodes with objects which it has not been trained to include in summaries. Table 3 displays a breakdown of zero-shot performance on long summaries. For comparison, results when nothing is held out—the standard case detailed in table 1—and for a model not trained to answer questions on the held out set are included. These comparisons show that while zero-shot summarization is not as accurate as fully supervised summarization, training on the auxilliary question-answering task is significantly better than not doing so. A model not trained to answer questions on episodes with held out objects is unable to correctly summarize episodes involving those held out objects. It is simply not able to output any of the held out objects’ names without having at least seen them during question answering. Training the model to learn to answer questions about the objects through an auxilliary question-answering task leads to clear improvement on the summarization task.

This result suggests that the model is learning representations of objects, or actions involving objects, while learning to answer questions which it can then use when producing summaries. There must be at least some transfer of representational knowledge between the question answering and the summarization tasks within the model.

Clear improvement with transfer compared to without transfer is also demonstrated in BLEU and ROUGE scores of both short and long summaries in seen and unseen environments (in only one case is there not improvement); see Table 5 in Appendix C for details.

3.2.1 Impact of question type on zero-shot transfer to summarization

We have seen that transfer from question answering to summarization occurs. But which questions are most important or useful for transfer? In order to further investigate the sharing of representations between question answering and summarization, we rerun the experiments using the same held out protocol, but using focused sets of particular question types. Testing each question type separately allows us to measure whether all questions are equally useful for promoting transfer to summarization.

Interestingly, we find that not all questions are equally useful: only the temporal “just before” and “just after” questions—which ask what action was performed just before or after a given action—exhibit transfer between tasks (see Table 3 for accuracy metrics on temporal and non-temporal questions). This is true of both subtypes of these questions, i.e. both the simple and complex language versions. On their own, the “yes/no” and “either/or” questions about objects or actions do not lead to the same zero-shot summarization ability. It is worth recalling here that the answers to the temporal questions were also found to be especially consistent with the long summaries in the missing object errors they contained, which would also suggest a particularly aligned representational space between these tasks (see Table 2).

Table 3 Accuracy of zero-shot long summarization when transferring representations learned from question answering to producing long summaries, broken down by question type used to learn the objects held out from summarization training.

Model trained on	Seen envs	Unseen envs
All questions	.519 ±.067	.302 ±.156
Temporal questions	.566 ±.118	.299 ±.085
All other non-temporal questions	.006 ±.007	.000 ±.000
No Q&A training on held out objects	.000 ±.000	.000 ±.000
All objects and questions - nothing held out	.850 ±.005	.475 ±.035

Results shown for episodes containing held-out objects in the validation sets in seen and unseen environments. The bottom two rows show a baseline with no question answering training on the held-out objects—and therefore no transfer—and a comparison to the fully trained model with nothing held out

Long Summaries

go to the ~~countertop~~ *(coffee machine)*, pick up the knife, go to the tomato, slice the tomato, put the knife in the garbage can, pick up the tomato, go to the bowl, put the tomato on the bowl, pick up the bowl, go to the fridge, put the bowl in the fridge

go to the ~~armchair~~ *(dining table)*, pick up the ~~pen~~ *(newspaper)*, go to the sofa, put the ~~pen~~ *(newspaper)* on the sofa

go to the ~~armchair~~ *(sofa)*, pick up the remote control, go to the coffee table, put the remote control on the coffee table, go to the remote control, pick up the remote control, go to the coffee table, put the remote control on the coffee table

go to the sink basin, pick up the ~~mug~~ *(cup)*, go to the sink basin, put the ~~mug~~ *(cup)* in the sink basin

Short Summaries

put a plate with a statue on it on the table *(put a plate with tissues on it onto the dining table)*

put a cooked slice of potato in the sink *(put a sliced egg inside the sink)*

examine a pillow by the light of a small lamp *(examine a pillow in the light of a tall lamp)*

Fig. 3 Example errors in generated long and short summaries. Errors in the long summaries are indicated with strikethrough text (with the correct text following in italics and parentheses). Generated short summaries appear to the left of the correct summaries, which are in italics

We also tested the transfer ability of a model trained in a similar manner but which excluded episodes based on the action verbs they contained rather than the objects. For these experiments, only one action verb at a time and the episodes which contained it were identified as held out items. In none of these cases was the model able to transfer the use of the verb to summaries of the held out episodes. This could be due to the smaller number of actions in the dataset than objects.

4 Related Work

RoboNLP Tangiuchi et al. (2019) and Tellex et al. (2020) offer thorough reviews of language use in the context of robotics. Detailed descriptions of actions such as robots playing soccer (Mooney, 2008) or automated driving (Barrett et al., 2015, 2017) have been generated. These have not involved learning how to report and condense a series of actions into anything like a summary, however. DeChant and Bauer (2021) propose robot action summarization as a research direction, suggesting a set of tasks to pursue.

Instruction Following Our proposal is closely related to learning to follow natural language instructions, which has long generated a great deal of interest at the intersection of robotics and natural language processing (Winograd, 1972; Dzifcak et al., 2009). Shridhar et al. (2021a) train a robotic arm in a virtual environment to perform a range of tasks following natural language instructions and transfer the learned model to a real world robot. Mees et al. (2021) introduce a benchmark for long horizon robotic manipulation tasks following natural language instructions.

Rich simulated environments for language-guided navigation tasks have been introduced in recent years. Anderson et al. (2018) introduced the Room to Room vision and language navigation dataset, which became the basis for much work in this area. Some of that work has involved learning to generate natural language descriptions of navigation trajectories as a training signal or tool: Nguyen et al. (2021) provide feedback to an agent in the Room to Room environment by describing in natural language the paths the agent actually takes so it can learn to compare that to the path it should have taken; Fried et al. (2018) learn to generate instructions to augment training data and then, at test time, to evaluate the similarity of routes it might take with the description of the desired route.

The ALFRED dataset (Shridhar et al., 2020) we repurpose has inspired a great deal of work on its natural language instruction following challenge. Shridhar et al. (2021b) improve an agent's ability to perform tasks in the virtual environment by first training them to learn to act in the interactive text only TextWorld environment (Côté et al., 2018) in similar situations which are described there only in text. Pashevich et al. (2021) learn to leverage the presence of the high level PDDL plans to produce better representations of the natural language instructions by also training those representations to be used to generate PDDL plans from the natural language instructions.

Q&A in robotics Learning to ask questions has also been worked on as a way for a robotic agent to ask for help or clarification while performing a task (Tellex et al., 2014; Thomason et al., 2019). Yoshino et al. (2021) use natural language questions to clarify aspects of how a simple action was performed in response to a question. Datta et al. (2022) introduce a form of question answering where the questions are in natural language but the answers take the form of visual highlights of a map to indicate locations. Carta et al. (2022) propose filling in the blanks within structured language instructions as an auxiliary task for reinforcement learning agents in a 2-D grid world. Gao et al. (2021) introduce a similar Q&A task in a virtual environment, though without summarization; a slightly different embodied Q&A task, requiring an agent to seek out answers to questions, is proposed by Gordon et al. (2018).

Summarization There is an extensive body of work on natural language summarization, providing examples and resources for the new but related task of robot action summarization (see Nenkova and McKeown (2012) and Gambhir and Gupta (2017) for reviews). There are two main kind of summarization. In extractive summarization, the summaries are selected from the original text already present in a source document. In abstractive summarization, by contrast, new text is generated as the summary, allowing for a higher level of description. Recurrent sequence to sequence models (Rush et al., 2015; Gupta & Gupta, 2019) as well as Transformer (Vaswani et al., 2017) models have been used to perform abstractive summarization (Lewis et al., 2019; Raffel et al., 2020).

Video understanding Work on understanding video is relevant to our work since we are interested in using video or selected images from video as an input to summarizing a robot's action in natural language. The task of 'video summarization' in the computer vision community refers to selecting important frames of a video that can, together, serve as a visual summary of the whole video; see Apostolidis et al. (2021) for a review of such techniques. Some work has been done on multimodal summarization from video and text transcripts to natural language summaries; Palaskar et al. (2019) is one example, going from video and text in the How2 video dataset (Sanabria et al., 2018) to summaries. Bärmann and Waibel (2022) assemble a large question answering dataset for real world video of humans performing actions, requiring significant effort to annotate.

Natural language question answering is also used for video understanding. Originally stemming from similar work in visual question answering (VQA) of natural language questions on still images (Antol et al., 2015), many video Q&A works address factual questions about the presence of objects or particular actions in video clips (Fan, 2019; Castro et al., 2022). These questions are similar to the object and action questions in our work. More recently, video question answering work has focused on more complex questions, including questions about the order of actions which are similar to our temporal questions (Xiao et al., 2021; Grunde-McLaughlin et al., 2021). Work has also been done to answer causal and related questions (e.g. "why did X happen?") which we do not address here and leave for future work (Wu et al., 2021; Li et al., 2022). Video question and answering has also been done with multimodal input which incorporates both video and at least one other modality such as text captions or an audio track (Choi et al., 2021; Yang et al., 2022). While our work does not incorporate such multimodal sources, future robot action summarization could do so, particularly for robots that have natural language interaction with humans in the course of their operation. Some video question answering datasets contain questions which are automatically formed from natural language descriptions of video sequences (Zeng et al., 2017; Zhao et al., 2017). Our automatic question generation method is similar but also incorporates ground truth information about the environment which are accessible because the episodes take place in simulation. Pretrained language models have been incorporated in models used to address video question answering (Zellers et al., 2021). See a recent survey by Zhong et al. (2022) for additional background on question answering for video understanding.

Grounding language It has been recognized for some time that grounding language to the real world is essential for creating AI systems that actually understand the language they processed (Harnad, 1990). Recent proposals on the need to situate natural language processing in a grounded or embodied context have brought renewed attention to this issue (Bisk et al., 2020; Chandu et al., 2021; McClelland et al., 2020; Lake & Murphy, 2021). Though these did not discuss robots summarizing their actions, our work is a contribution to this direction of research.

5 Conclusion

We develop a model that can be jointly trained to summarize and answer questions about a virtual robot's past actions. We find that the model learns a representation space which is shared across at least some of the question types and summaries, leading to zero-shot summarization abilities.

This work helps begin a line of research on robot action summarization and question answering. It is important that robots operating in the real world be well supervised by humans and that their actions be understandable. We suggest that establishing a basic narrative of *what* an agent does is in some ways a prerequisite to further understanding *why* it does something. Once answering questions about and summarizing robot actions can be performed reliably, we expect these capabilities to be useful in a variety of ways, including in training robots. Learning representations for these tasks can serve as a form of pretraining for downstream robotic applications. New techniques for lifelong learning might enable robots to receive and learn from feedback to the summaries they generate. Our approach of making use of an existing instruction following dataset naturally allows for this application and is something we will pursue in future work on this and other datasets.

Though this work took place in simulation, the summarization and question answering tasks are not specific to aspects of this or any simulated environment. Future work will explore the application of these tasks to real world robots.

Author Contributions C.D. led the project, performed experiments, and wrote the initial draft of the manuscript. I.A. and D.B. provided supervision, research guidance, and edited and wrote portions of the manuscript.

Funding C.D. was supported by a grant from the Long Term Future Fund

Code availability Code and data will be released upon publication

Declarations

Conflict of interest The authors declare no conflict of interest

Ethical approval The work performed for this paper made no use of human subjects.

Appendix A Neural network model and training details

A.1 Neural network

We train a bridge network to downsize and link the output of the last convolutional layer of a pretrained CLIP Resnet-50 network (Radford et al., 2021) with a pretrained T5 transformer ("t5-base") from the Hugging Face library (Raffel et al., 2020; Wolf et al., 2020).

Bridge network architecture

Input: 2048 × 7 × 7 Resnet features

Conv layer 1 (2048, 1024, 1)

Conv layer 2 (1024, 128, 1)

Conv layer 3 (128, 32, 1)

Fully connected layer (1568, 768)

Total number of parameters

The CLIP Resnet-50 network has a total of 102,007,137 parameters.

The bridge network described above has a total of 3440,864 parameters.

The T5 Transformer network we fine-tune has a total of 222,903,552 parameters.

The bridge network and T5 networks together (the complete model we train / fine-tune) have a total of 226,344,416 parameters.

A.2 Training information

We primarily used two Titan Xp and three NVIDIA A6000 GPUS. When training one of our models with all questions and training data, a Titan Xp took approximately 3 days and an A6000 approximately 1.5 days to train for 100 epochs. Though we did not track it, a rough estimate of total GPU time during initial exploration of this problem and the work reported here is 5000 h.

Hyperparmeters we tested variations of include the optimizer (Adam, AdamW, AdaFactor, AdamW—Adam was used in all experiments reported here); the learning rate (.001 was used as the initial learning rate in all experiments reported here); and network architecture choices for the bridge network which connected the CLIP Resnet convolutional layer outputs and the T5 transformer (layer sizes, number of layers, batchnorm, dropout). The random seed was not one of the hyperparameters tuned; we used three random seeds to produce all of the results in the paper, which are averaged across three runs with different random seeds and, in the case of the experiments involving held out objects, three randomly chosen sets of five objects.

We use the dataset's valid seen set as our validation set with which to choose hyperparameters and select the epoch for results to report. We use the accuracy metric of the long summarization task as the measure to select the best epoch. We then report the results of short summarization and all questions in that epoch. These are not the best epochs reached for each of the questions but we report the results from a single epoch to be consistent. The epoch we report best results from the valid seen set is also rarely the best epoch for the valid unseen set but we report the results for the valid unseen set from the same epoch.

Our automatic precision, BLEU, and ROUGE metric was generated from an implementation available through the Hugging Face library.

The ALFRED dataset was released under an MIT License and we release our questions and answers dataset under the same license.

Appendix B Ablation of visual input

To investigate how the model would perform on summarization and question answering tasks in the absence of any meaningful input, we trained the model as usual but instead of presenting a sequence of images corresponding to each episode, we presented only one sequence of images from one episode for every summarization prompt and question. Therefore no useful information about each episode was input to the model. The output on validation set summarization and question answering prompts would therefore only be a reflection of what the model has learned about the regularities in the text portion of the dataset, e.g. what actions are more likely to follow from other actions, regardless of the episode visual data. Table 4 presents the results of the ablation study on the seen and unseen validation set environments.

Some of the binary questions have accuracies very close to 50%—these include the “simple action yes/no”, “complex action yes/no”, and “simple action yes/no”. The “object yes/no” and “object either/or” questions, on the other hand, have slightly higher accuracies, approximately 63%, suggesting that model has learned some patterns in the distribution of objects in the dataset. Similarly, the temporal “just before” and “just after” questions have higher accuracies than a uniformly random choice among possible actions would

Table 4 Ablation of video frames baseline: results for a model trained to answer questions and produce summaries when trained with questions and answers as usual but with each question and answer pair and summarization task paired to identical visual input (i.e. each episode’s observations are replaced by a single, static set of observations that do not vary from episode to episode), thereby completely depriving the model of any useful visual information with which to answer the question

Question / prompt	Seen envs accuracy	precision	Unseen envs accuracy	precision
Object yes/no	.629 ±.034	–	.636 ±.030	–
Object either/or	.632 ±.010	.636 ±.034	.635 ±.008	.640 ±.012
Simple action yes/no	.503 ±.029	–	.533 ±.038	–
Complex action yes/no	.486 ±.010	–	.508 ±.032	–
Simple action either/or	.526 ±.028	.710 ±.001	.535 ±.017	.725 ±.015
Simple action just before	.479 ±.016	.752 ±.005	.517 ±.007	.775 ±.014
Complex action just before	.539 ±.007	.766 ±.004	.613 ±.025	.826 ±.017
Simple action just after	.315 ±.022	.706 ±.017	.349 ±.039	.728 ±.021
Complex action just after	.256 ±.018	.658 ±.003	.325 ±.019	.695 ±.003
Long summary	.005 ±.002	.492 ±.094	.000 ±.000	.471 ±.101
	ROUGE	Precision	ROUGE	Precision
Short summary	.264 ±.027	.496 ±.058	.263 ±.022	.485 ±.059

Table 5 ROUGE and BLEU scores of zero-shot long summarization when transferring representations learned from question answering to producing long (at the top) and short (at the bottom) summaries, broken down by question type used to learn the objects held out from summarization training.

Question / prompt	Seen envs ROUGE	BLEU	Unseen envs ROUGE	BLEU
Long summaries				
All questions	.919 ±.007	.888 ±.018	.866 ±.006	.813 ±.068
Temporal questions	.940 ±.004	.910 ±.035	.858 ±.005	.798 ±.004
All other (i.e. non-temporal) questions	.810 ±.003	.697 ±.026	.765 ±.008	.651 ±.088
No Q&A training on held out objects	.802 ±.004	.688 ±.022	.806 ±.003	.703 ±.022
All objects and questions - nothing held out	.981 ±.000	.969 ±.000	.922 ±.004	.880 ±.010
Short summaries				
All questions	.483 ±.002	.370 ±.049	.412 ±.005	.314 ±.100
Temporal questions	.465 ±.003	.378 ±.040	.421 ±.006	.347 ±.091
All other (i.e. non-temporal) questions	.404 ±.003	.276 ±.041	.365 ±.005	.227 ±.052
No Q&A training on held out objects	.401 ±.003	.277 ±.045	.427 ±.003	.288 ±.049
All objects and questions—nothing held out	.571 ±.000	.556 ±.006	.517 ±.004	.504 ±.022

Results shown for episodes containing held-out objects in the validation sets in unseen and seen environments. The bottom two rows of the long and short summary sections show baselines with no question answering training on the held-out objects—and therefore no transfer—and a comparison to the fully trained model with nothing held out. This table complements Table 3 in the main body of the paper, which provides binary all-or-nothing accuracy scores for long summaries demonstrate. Our actual model achieves much higher accuracy on all of these tasks, however, demonstrating that it has learned much more than merely the regularities in the dataset.

We initially tested an additional form of question, a question that asked if <Action A> happened before <Action B> in a given episode (note that this type of question differs from the temporal questions included in this work, which require the model to output the action that happened *immediately* before or after a given action, not a binary indicator of whether an action happened at any point before a given action). We had to exclude this form of question, however, because the model was able to achieve over 80% accuracy on validation set episodes under the ablated visual input regime. This question was apparently simply too easy given the regularities in the dataset.

Appendix C Additional comparison metrics for zero-shot transfer

In Table 5 we provide additional metrics to understand the performance of zero-shot transfer from question answering to summarization. Short summaries tend to have lower BLEU and ROUGE scores than the long summaries because the long summaries use a standardized set of words to describe actions and objects while the short summaries use a more diverse set of words and provide varying levels of detail.

Appendix D Out of distribution negative questions

Our model was trained on a set of questions which involved a certain set of actions and objects in a particular (household) environment. What would the model do if asked questions about actions it did not engage in with objects it had not seen, either at test time or during training? This would be a particular issue for yes/no questions if an agent were asked if it had engaged in an action it did not engage in and was not familiar with, as it may be expected to respond essentially randomly to such unfamiliar questions.

To begin to investigate how such a model deals with these out of distribution negative questions, we developed two small test sets of questions:

Table 6 Accuracy of one trained model on small test sets of out of distribution negative questions, by question and environment type

Question type	Val seen	Val unseen
Ordinary Qs	.83	.86
Extraordinary Qs	.64	.76

1. Ordinary Out of Distribution Questions, consisting of questions about actions a household robot may be expected to take but which are not present in our dataset in any form, such as:

"did you clean the attic?"
"did you move the toys?"
"did you do the laundry?"
"did you water the plants?"
"did you take out the garbage?"

2. Extraordinary Out of Distribution Questions, consisting of questions totally unrelated to robots or household chores, such as:

”did you swim to the coral reef?"
"did you learn German?"
"did you fall in love?"
"did you kayak in the fjord?"
"did you graduate from college?"

Both test sets consist of fifty such questions (released along with our larger set of questions and answers paired to the ALFRED dataset). The correct answer to all of the questions is "no". The questions were run through a fully trained model (no held out objects). Results are shown in Table 6.

The model generally demonstrates a (in this case, correct) bias for answering "no" to these out of distribution questions. Perhaps surprisingly, the model is much more likely to correctly answer "no" to *Ordinary* Out of Distribution Questions than it is to *Extraordinary* Out of Distribution Questions.

These results, though on a small test set, suggest that the model has learned a bias toward answering, "yes", only when there is evidence in the input that an answer should be answered affirmatively. This is, of course, what a user would want. Further investigation of the circumstances under which it correctly answers out of distribution questions is warranted, as well as ways to improve the performance on out of distribution questions, especially unusual ones.

Appendix E Fine tuning with LoRA

In all experiments we present in the main body of the paper we fine-tune a pretrained T5 LLM as well as train the bridge network from scratch. In order to investigate whether fine-tuning the entire network is necessary, we also used the LoRA technique proposed by Hu et al. (2021). LoRA allows for more efficient fine-tuning because it allows for freezing a model's original weights while only learning a much smaller number of parameters in rank decomposition matrices attached to layers of the original model. In this case, the number of trainable parameters associate with the T5 model is reduced by 99.6%.

While training with LoRA is more efficient, we found that, in all but one metric in one task, it reduces performance in the tasks we train our model on, as can be seen in Table 7. In

Table 7 Performance of our multimodal model when its T5 transformer is trained using the LoRA method by question or prompt.

Question / prompt	Seen envs accuracy	difference	Unseen envs accuracy	difference
Object yes/no	.943 ±.02	−.011	.885 ±.005	−.022
Object either/or	.980 ±.009	−.01	.950 ±.005	−.16
Simple action yes/no	.965 ±.008	−.010	.846 ±.006	−.046
Complex action yes/no	.922 ±.017	−.013	.877 ±.004	−.018
Simple action either/or	.981 ±.004	−.007	.906 ±.013	−.017
Simple action just before	.932 ±.015	−.016	.761 ±.019	−.104
Complex action just before	.911 ±.009	−.016	.751 ±.006	−.067
Simple action just after	.956 ±.011	−.003	.732 ±.007	−.083
Complex action just after	.886 ±.015	−.025	.671 ±.021	−.059
Long Summary	.754 ±.010	−.096	.394 ±.014	−.081
	ROUGE	Difference	ROUGE	Difference
Short Summary	.571 ±.010	−.012	.500 ±.010	−.017
	BLEU	Difference	BLEU	Difference
Short Summary	.981 ±.013	.009	.461 ±.013	−.043

Performance and the difference between training with LoRA and training with fine-tuning the entire T5 (See Table 1) is shown for seen and unseen environment validation sets

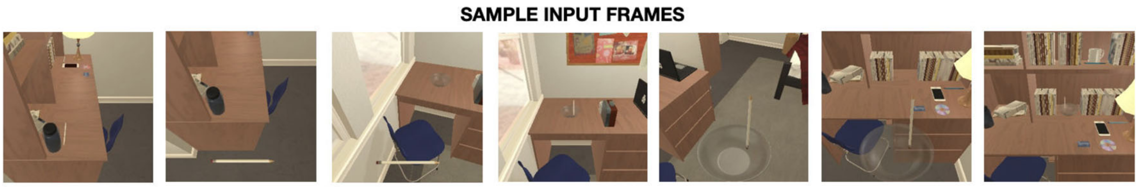
some cases, particularly the long summary task, the degradation in performance is dramatic, showing a decline of 9.6 and 8.1 percentage points in seen and unseen environments, respectively. On many of the question and answer tasks the difference is rather minor in the seen environments but, interestingly, much more significant in the unseen environments. Only in the case of the BLEU score for the short summary task in seen environments do we see a very small increase when training with LoRA.

Further investigation is required to determine why LoRA underperforms in some tasks rather than others and, why it especially underperforms in unseen environments requiring more generalization ability, and, more generally, why its performance is nearly uniformly worse than fine-tuning the entire T5 model. We hypothesize that our method, which involves converting a uni-modal text-only LLM to process representations of video frames alongside text, benefits from being fully fine-tuned because of the magnitude of the domain change seen in its inputs.

We use the implementation of LoRA in the huggingface PEFT (Mangrulkar et al., 2022) library with the settings found in that library specified for the T5 model.

Appendix F Additional episode example

See Fig. 4 for examples of a selection of video frames and questions from an episode in the valid unseen set.



Question / Prompt	Answer
OBJECT QUESTIONS	
Was there a glass bottle?	no
Was there a candle or a safe?	safe
SIMPLE ACTION QUESTIONS	
Did you go to the bowl?	yes
Did you go to the desk or go to the fridge?	go to the desk
COMPLEX ACTION QUESTIONS	
Did you go across the room, then turn left and go toward the wall, then turn left to face the desk?	yes
SIMPLE TEMPORAL QUESTIONS	
What did you do just before go to the shelf?	pick up the bowl
What did you do just after put the pencil on the bowl?	pick up the bowl
COMPLEX TEMPORAL QUESTIONS	
What did you do just before turn around and move forward to the bed, then turn right and go to the desk ahead?	pick up the pencil
What did you do just after turn to the left go right turn to the left go right turn to the right towards the desk?	put the pencil on the bowl
LONG SUMMARIES	
Narrate what you did.	go to the desk, pick up the pencil, go to the bowl, put the pencil on the bowl, pick up the bowl, go to the shelf, put the bowl on the shelf
SHORT SUMMARIES	
Summarize what you did.	collect pencil and bowl for the desk

Fig. 4 Sample partial selection of input frames from an episode in an unseen environment originally from the ALFRED dataset (at the top), generated questions (on the left, in blue) and expected answers (on the right, in green), broken up into question type, along with the prompts for long and short summaries, at the bottom

References

Anderson, P., Wu, Q., Teney, D., et al. (2018). Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3674–3683.

Antol, S., Agrawal, A., Lu, J., et al. (2015). Vqa: Visual question answering. In Proceedings of the IEEE international conference on computer vision, pp. 2425–2433.

Apostolidis, E., Adamantidou, E., Metsai, A.I., et al. (2021). Video summarization using deep neural networks: A survey. arXiv preprint arXiv:2101.06072.

Bärmann, L., & Waibel, A. (2022). Where did i leave my keys? - episodic-memory-based question answering on egocentric videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp 1560–1568.

Barrett, D.P., Bronikowski, S.A., Yu, H., et al. (2015). Robot language learning, generation, and comprehension. arXiv preprint arXiv:1508.06161.

Barrett, D. P., Bronikowski, S. A., Yu, H., et al. (2017). Driving under the influence (of language). *IEEE Transactions on Neural Networks and Learning Systems, 29*(7), 2668–2683.

Bisk, Y., Holtzman, A., Thomason, J., et al. (2020). Experience grounds language. arXiv preprint arXiv:2004.10151.

Carta, T., Lamprier, S., Oudeyer, P. Y., et al. (2022). Eager: Asking and answering questions for automatic reward shaping in language-guided rl. arXiv preprint arXiv:2206.09674.

Castro, S., Deng, N., Huang, P., et al. (2022). In-the-wild video question answering. In Proceedings of the 29th International Conference on Computational Linguistics, pp 5613–5635.

Chandu, K. R., Bisk, Y., Black, A. W. (2021). Grounding'grounding'in nlp. arXiv preprint arXiv:2106.02192.

Choi, S., On, K. W., Heo, Y. J., et al. (2021). Dramaqa: Character-centered video story understanding with hierarchical qa. In Proceedings of the AAAI Conference on Artificial Intelligence, pp 1166–1174.

Côté, M.A., Kádár, A., Yuan, X., et al. (2018). Textworld: A learning environment for text-based games. In Workshop on Computer Games, Springer, pp 41–75.

Datta, S., Dharur, S., Cartillier, V., et al. (2022). Episodic memory question answering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 19119–19128.

DeChant, C., & Bauer, D. (2021). Toward robots that learn to summarize their actions in natural language: a set of tasks. In: 5th Annual Conference on Robot Learning, Blue Sky Submission Track.

Dzifcak, J., Scheutz, M., Baral, C., et al. (2009). What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In 2009 IEEE International Conference on Robotics and Automation, IEEE, pp 4163–4168.

Fan, C. (2019). Egovqa-an egocentric video question answering benchmark dataset. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp 0–0.

Fried, D., Hu, R., Cirik, V., et al. (2018). Speaker-follower models for vision-and-language navigation. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp 3318–3329.

Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review, 47*(1), 1–66.

Gao, D., Wang, R., Bai, Z., et al. (2021). Env-qa: A video question answering benchmark for comprehensive understanding of dynamic environments. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 1675–1685

Gordon, D., Kembhavi, A., Rastegari, M., et al. (2018). Iqa: Visual question answering in interactive environments. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4089–4098.

Grunde-McLaughlin, M., Krishna, R., & Agrawala, M. (2021). Agqa: A benchmark for compositional spatio-temporal reasoning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 11287–11297.

Gupta, S., & Gupta, S. K. (2019). Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications, 121*, 49–65.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena, 42*(1–3), 335–346.

He, K., Zhang, X., Ren, S., et al. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Hu, E. J., Shen, Y., Wallis, P., et al. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Kolve, E., Mottaghi, R., Han, W., et al. (2017). Ai2-thor: An interactive 3d environment for visual ai. arXiv preprint arXiv:1712.05474.

Lake, B. M., & Murphy, G. L. (2021). Word meaning in minds and machines. Psychological Review.

Lewis, M., Liu, Y., Goyal, N., et al. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

Li, J., Niu, L., & Zhang, L. (2022). From representation to reasoning: Towards both evidence and commonsense reasoning for video question-answering. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 21273–21282.

Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. In Text summarization branches out, pp. 74–81.

Lu, K., Grover, A., Abbeel, P., et al. (2022). Frozen pretrained transformers as universal computation engines. In Proceedings of the AAAI conference on artificial intelligence, pp. 7628–7636.

Mangrulkar, S., Gugger, S., Debut, L., et al. (2022). Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

McClelland, J. L., Hill, F., Rudolph, M., et al. (2020). Placing language in an integrated understanding system: Next steps toward human-level performance in neural language models. *Proceedings of the National Academy of Sciences, 117*(42), 25966–25974.

McDermott, D., Ghallab, M., Howe, A., et al. (1998). Pddl-the planning domain definition language. Technical Report, Tech Rep.

Mees, O., Hermann, L., Rosete-Beas, E., et al. (2021). Calvin—a benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. arXiv preprint arXiv:2112.03227.

Mooney, R. J. (2008). Learning to connect language and perception. In AAAI, pp. 1598–1601.

Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In Mining text data. Springer, pp. 43–76.

Nguyen, K. X., Misra, D., Schapire, R., et al. (2021). Interactive learning from activity description. In International conference on machine learning, PMLR, pp. 8096–8108.

Palaskar, S., Libovický, J., Gella, S., et al. (2019). Multimodal abstractive summarization for how2 videos. In ACL.

Papineni, K., Roukos, S., Ward, T., et al. (2002). Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the association for computational linguistics, pp. 311–318.

Pashevich, A., Schmid, C., & Sun, C. (2021). Episodic transformer for vision-and-language navigation. arXiv preprint arXiv:2105.06453.

Radford, A., Kim, J. W., Hallacy, C., et al. (2021). Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020.

Raffel, C., Shazeer, N., Roberts, A., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research, 21*, 1–67.

Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. In Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 379–389.

Sanabria, R., Caglayan, O., Palaskar, S., et al. (2018). How2: A large-scale dataset for multimodal language understanding. ArXiv arXiv:1811.00347.

Shridhar, M., Thomason, J., Gordon, D., et al. (2020). Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10740–10749.

Shridhar, M., Manuelli, L., & Fox, D. (2021a). Cliport: What and where pathways for robotic manipulation. In 5th Annual conference on robot learning.

Shridhar, M., Yuan, X., Cote, M. A., et al. (2021b). ALFWorld: Aligning text and embodied environments for interactive learning. In International conference on learning representations, https://openreview.net/forum?id=0IOX0YcCdTn.

Tangiuchi, T., Mochihashi, D., Nagai, T., et al. (2019). Survey on frontiers of language and robotics. *Advanced Robotics, 33*(15–16), 700–730.

Tellex, S., Knepper, R., Li, A., et al. (2014). Asking for help using inverse semantics. Robotics: Science and Systems X.

Tellex, S., Gopalan, N., Kress-Gazit, H., et al. (2020). Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems, 3*, 25–55.

Thomason, J., Padmakumar, A., Sinapov, J., et al. (2019). Improving grounded natural language understanding through human-robot dialog. In 2019 International conference on robotics and automation (ICRA), IEEE, pp. 6934–6941.

Tsimpoukelli, M., Menick, J. L., Cabi, S., et al. (2021). Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems, 34*, 200–212.

Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. In Advances in neural information processing systems, pp. 5998–6008.

Winograd, T. (1972). Understanding natural language. *Cognitive Psychology, 3*(1), 1–191.

Wolf, T., Chaumond, J., Debut, L., et al. (2020). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, pp. 38–45.

Wu, B., Yu, S., Chen, Z., et al. (2021). Star: A benchmark for situated reasoning in real-world videos. In Thirty-fifth conference on neural information processing systems datasets and benchmarks track (Round 2).

Xiao, J., Shang, X., Yao, A., et al. (2021). Next-qa: Next phase of question-answering to explaining temporal actions. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9777–9786.

Yang, P., Wang, X., Duan, X., et al. (2022). Avqa: A dataset for audio-visual question answering on videos. In Proceedings of the 30th ACM international conference on multimedia, pp. 3480–3491.

Yoshino, K., Wakimoto, K., Nishimura, Y., et al. (2021). Caption generation of robot behaviors based on unsupervised learning of action segments. In Conversational dialogue systems for the next decade. Springer, pp. 227–241.

Zellers, R., Lu, X., Hessel, J., et al. (2021). Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems, 34*, 23634–23651.

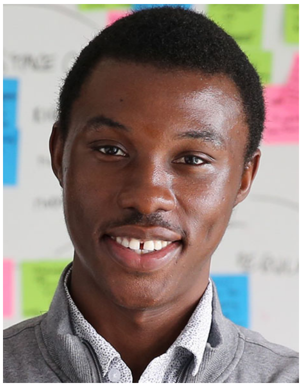
Zeng, K. H., Chen, T. H., Chuang, C. Y., et al. (2017). Leveraging video descriptions to learn video question answering. In Proceedings of the AAAI conference on artificial intelligence.

Zhao, Z., Lin, J., Jiang, X., et al. (2017). Video question answering via hierarchical dual-level attention network learning. In Proceedings of the 25th ACM international conference on Multimedia, pp. 1050–1058.

Zhong, Y., Ji, W., Xiao, J., et al. (2022). Video question answering: Datasets, algorithms and challenges. In Proceedings of the 2022 conference on empirical methods in natural language processing. association for computational linguistics, Abu Dhabi, United Arab Emirates, pp. 6439–6455, https://aclanthology.org/2022.emnlp-main.432.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Chad DeChant is a Ph.D. candidate in the Computer Science department at Columbia University. He received an M.A. in Government and B.A. in Philosophy from Georgetown University. His research interests include the intersection of Natural Language Processing and Robotics as well as how to ensure that robots and other AI-enabled systems operate safely and reliably.



Iretiayo Akinola is a Research Scientist at NVIDIA. He received a Ph.D. from Columbia University's Computer Science department in 2021, where his research centered on equipping robots with multi-modal (visual and tactile) perception to enhance self-awareness and reactive manipulation of objects. Prior to that, he earned his M.S. and BSc. degrees in Electrical Engineering from Stanford University and Obafemi Awolowo University, respectively. His research interests include multi-modal robot learning, human-robot interaction, and human-in-the-loop robot learning.

Daniel Bauer is a lecturer in the Computer Science department at Columbia University. He received his Ph.D. from Columbia in 2017, an M.Sc. in language science and technology from Saarland University (Germany) in 2009 and a B.Sc. in Cognitive Science from the University of Osnabrück (Germany). His research interests in Natural Language Processing include lexical and computational semantics and multimodal NLP.