

Do you see what I see?

Using questions and answers to align representations of robotic actions

Chad DeChant

Computer Science Department
Columbia University
chad.dechant@columbia.edu

Iretiayo Akinola

NVIDIA Corporation
iakinola@nvidia.com

Daniel Bauer

Computer Science Department
Columbia University
bauer@cs.columbia.edu

Abstract: We introduce the task of learning to answer questions about a robotic agent’s past actions using natural language alone. Humans represent much of our experience using language so training agents to answer questions about their experiences is one way of ensuring that humans and robotic agents see the world in similar ways. We develop a method to automatically generate questions and answers about objects, actions, and the temporal order in which they appeared in particular episodes by making use of an existing dataset and virtual environment. We then combine a convolutional network with a pretrained language transformer model to answer the questions from multimodal input, viz. questions embedded in the transformer model’s embedding space concatenated with egocentric video frame features. The model achieves a high level of accuracy (90%+ accuracy on thirteen of eighteen evaluation sets).

Keywords: RoboNLP, Language, Representations

1 Introduction

Human interaction depends on a shared representation of at least some aspects of the world and our experience of it. When teaching children or training people to perform a new skill we often have to explicitly encourage others to represent the world the way we want them to and take steps to find out if they are doing so. It is common to use language to do this. A human might ask questions about what someone else has done and seen in order to better understand the second person’s actions and to make sure that they share a common understanding of the situation.

When robots operate in environments together with humans we would similarly like to ensure that their representations of the world are aligned with those of the humans they operate alongside. As in the case of human to human communication, we suggest that this can be done in part using language. To that end we propose the task of enabling robotic agents to learn to answer questions about their past actions in language. We develop a method to produce sets of questions and answers either in advance of training or on the fly during training. Derived from simulator metadata, our approach is more efficient than having to rely on human-generated questions and answers. In order to answer the generated questions about episodes, we train a model that combines a frozen pretrained vision network, a pretrained language transformer model which we fine-tune, and a convolutional network trained from scratch to bridge the two pretrained networks.

We hypothesize that question answering can be an effective way to improve the alignment of representations obtained by robots during task learning with the representations humans have about the world. To understand robot interaction in an environment, three aspects of an episode stand out as being especially important: the objects present in the environment; the actions taken by the agent;





		OBJECT QUESTIONS	
		Was there a fork?	yes
		Was there or fridge or a safe?	fridge
		ACTION QUESTIONS	
		Did you gotolocation desk?	no
		Did you sliceobject apple or washobject apple?	sliceobject apple
		TEMPORAL QUESTIONS	
		Did you sliceobject apple before gotolocation fridge?	yes
		Which did you do first, gotolocation fridge or gotolocation garbagecan?	gotolocation fridge
		What did you do just before gotolocation garbagecan?	coolobject apple
		What did you do just after putobject knife fridge?	gotolocation sinkbasin
FULL NARRATIVE QUESTION			
What did you do?		gotolocation countertop pickupobject knife gotolocation apple sliceobject apple gotolocation fridge putobject knife fridge gotolocation sinkbasin pickupobject apple gotolocation fridge coolobject apple gotolocation garbagecan putobject apple garbagecan	

Figure 1: Sample questions (in blue) and answers (in green), broken up into question type, along with a selection of video frames (in clockwise direction) of the corresponding episode in the dataset.

and the temporal sequence in which the episode unfolded. Generating questions targeted at these aspects can be a way to guide representations learned by robots.

2 Method

Our approach requires egocentric video, 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 [1] dataset, making use of the preselected set of egocentric video frames and a description of the agent’s actions in the semi-structured Planning Domain Description Language (PDDL) [2]. We use the AI2THOR environment to rerun the agent trajectories present in the dataset and capture the metadata present while the agent is in the environment. This metadata includes information about objects encountered in the virtual environment and the order in which the robot sees or interacts with them. Though we use one particular existing dataset and environment, our approach can be used in other cases where similar data can be captured.

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. The algorithm can be used as a one-time offline dataset generation step or in an online fashion during training. The algorithm produces eight types of questions in three broad categories:

- (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 a <object>?”, which require only “yes” or “no” answers and “object either/or” questions of the form, “was there a <object A> or <object B>?” which require the model to output the name of the object present (only one of the objects will be present in the episode).
- (2) **action questions**, which ask about actions the agent performed. The two kinds of question — “action yes/no” and “action either/or” — follow the structure of the respective object questions.
- (3) **temporal questions** about the order in which actions were performed, of four kinds. The first kind, “action before yes/no” of the form, “did you <action A> before <action B>?”, are answered with a “yes” or “no”. The other three types are answered with an action as phrased in PDDL format followed by one or two object and/or place names in natural English: “action before either/or” questions (“which did you do first, <action A> or <action B>?”); “what action just before” questions (“what did you do just before <action A>?”); and “what action just after” questions (“what did you do just after <action A>?”).

In addition to the above types of questions, we use the full PDDL description of the agent’s actions as the answer to a ninth question, “what did you do?”.

Question	Seen envs		Unseen envs	
	Accuracy	Precision	Accuracy	Precision
Object yes/no	.935 \pm .008	-	.900 \pm .039	-
Object either/or	.969 \pm .010	.969 \pm .010	.940 \pm .022	.940 \pm .022
Action yes/no	.973 \pm .004	-	.885 \pm .007	-
Action either/or	.993 \pm .002	.996 \pm .002	.935 \pm .012	.953 \pm .011
Action before yes/no	.992 \pm .003	-	.969 \pm .007	-
Action before either/or	.983 \pm .003	.988 \pm .002	.982 \pm .005	.986 \pm .006
What action just before	.942 \pm .003	.992 \pm .003	.847 \pm .027	.918 \pm .027
What action just after	.944 \pm .008	.969 \pm .006	.811 \pm .014	.902 \pm .005
Full narration	.819 \pm .008	.969 \pm .011	.477 \pm .027	.882 \pm .032

Table 1: Accuracy and precision scores for answered questions by question type, averaged across three runs, including standard deviation. 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 not found in the training set. Precision scores not shown for “yes/no” answers where such scores must equal the accuracy scores.

Examples of each type of question can be found in Figure 1. Our Q&A generation algorithm samples the negative examples in the ‘yes/no’ and ‘either/or’ questions in proportion to their appearance in the training set so that the model cannot learn to answer based on statistics of the training set while ignoring the actual episodes. The algorithm also excludes temporal questions that could be ambiguous, which may be the case if an action is repeated in an episode. Not every episode, therefore, has questions of every type.

Question Answering Model

Here, we present a learned algorithm that takes as input egocentric videos frames of a virtual mobile robot and responds to questions about the robot’s interactions in the frames. Our full neural network model (see Figure 2) combines several components. Video frames are fed into a frozen Resnet network pretrained as part of the CLIP model. 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 T5 transformer model [3] (“t5-base” in the Hugging Face library [4]) which we fine-tune. While the T5 model was pretrained exclusively on language data, we use it both for language and visual input, following other work which has shown the ability of language model transformers to process multimodal data [5]. The natural language questions are embedded using the model’s pretrained embeddings. We concatenate the text embeddings with the image vector representations yielded by the bridge network. As the T5 is an encoder-decoder model it is able to generate encoding representations of the images conditioned on the question being asked. We train one model to answer all questions so it must learn to generate representations useful for all questions. During an epoch of training, one question of each type is asked for each episode (when such a question exists).

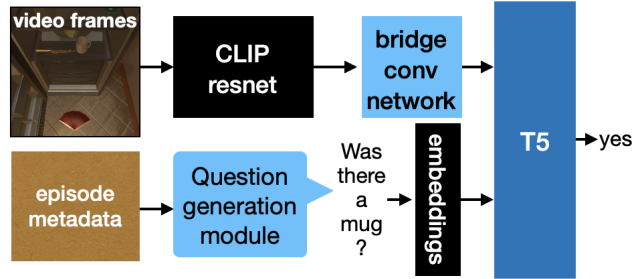


Figure 2: Input (at the left) to our full model includes video frames as well as episode metadata describing the environment as the agent saw it. The components in black are pre-trained and remain frozen during our training process, while the light blue modules are trained from scratch. The dark blue module, a pretrained T5, was finetuned during training.

3 Results

We find that our model performs very well on the questions from our Q&A generation algorithm. Table 1 presents accuracy results for all question types, and precision scores for questions that require generating text. An answer is considered accurate if it completely matches the target answer; precision measures what percentage of words in an output answer are in the target answer. 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 and whether the model was given the answer as a choice, i.e. as part of an “either/or” question. Providing the model a sort of hint in the form of an either/or question is an easier task than requiring the model to come up with the answer without that hint. By contrast, with longer answers there are more opportunities for errors so the worse the performance when measured by the standard of complete accuracy tends to be. This is particularly true for the question which asks for a full narration of the agent’s action, which has by far the worst results. For that question we also calculated BLEU scores to better characterize performance, finding an average BLEU score of 0.936 ± 0.013 in the previously seen environments and 0.775 ± 0.032 in the unseen environments.

Second, with only one exception, “either/or” questions see 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 make sense of than those about the mere occurrence of those actions. Surprisingly, then, we find that in most cases the model actually performs slightly better when asked about the order of actions than about their simple presence or absence in an episode.

The model tends to make two kinds of errors when generating anything other than “yes/no” answers. It sometimes misidentifies objects, particularly 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.

4 Related work

This work is in some ways performing the inverse operation of two common tasks at the interface of language and robotics. Learning to follow natural language instructions has a long history, from Winograd [6] to an abundance of recent work in virtual environments [7] and real world manipulation [8, 9]. 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 [10, 11].

Yoshino et al. [12] use natural language questions to clarify aspects of how a simple action was performed in response to a question. Datta et al. [13] 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. Bärmann and Waibel [14] assemble a large question answering dataset for real world video of humans performing actions, requiring significant effort to annotate. Carta et al. [15] propose filling in the blanks within structured language instructions as an auxiliary task for reinforcement learning agents in a 2-D grid world. Learning to produce instructions has been used to augment datasets and as a training signal [16, 17].

5 Conclusion

We have demonstrated a system for answering questions about robotic agents’ past actions in a virtual environment. Training an agent to answer such questions encourages it to represent the world

in ways similar to the ways humans represent it. Answering the kinds of questions we propose requires an agent to break up the world into the same objects and actions as humans do as well as understand their temporal relationship. This alignment of representations will help make robotic behaviors more understandable to humans. Unifying the representations used to make action decisions with those induced by our question answering task should also make natural language instructions more understandable to the robotic agents, facilitating the learning of complex tasks describable in language.

References

- [1] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.
- [2] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. Pddl-the planning domain definition language. 1998.
- [3] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.
- [4] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.
- [5] K. Lu, A. Grover, P. Abbeel, and I. Mordatch. Frozen pretrained transformers as universal computation engines. 2022.
- [6] T. Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.
- [7] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025, 2022.
- [8] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022.
- [9] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.
- [10] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy. Asking for help using inverse semantics. 2014.
- [11] J. Thomason, A. Padmakumar, J. Sinapov, N. Walker, Y. Jiang, H. Yedidsion, J. Hart, P. Stone, and R. J. Mooney. Improving grounded natural language understanding through human-robot dialog. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6934–6941. IEEE, 2019.
- [12] K. Yoshino, K. Wakimoto, Y. Nishimura, and S. Nakamura. Caption generation of robot behaviors based on unsupervised learning of action segments. In *Conversational Dialogue Systems for the Next Decade*, pages 227–241. Springer, 2021.
- [13] S. Datta, S. Dharur, V. Cartillier, R. Desai, M. Khanna, D. Batra, and D. Parikh. Episodic memory question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19119–19128, 2022.
- [14] L. Bärman and A. Waibel. 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*, pages 1560–1568, June 2022.
- [15] T. Carta, S. Lamprier, P.-Y. Oudeyer, and O. Sigaud. Eager: Asking and answering questions for automatic reward shaping in language-guided rl. *arXiv preprint arXiv:2206.09674*, 2022.

- [16] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell. Speaker-follower models for vision-and-language navigation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3318–3329, 2018.
- [17] M. T. Nguyen, P. T. Nguyen, V. V. Nguyen, and M. C. N. Hoang. Iterative multilingual neural machine translation for less-common and zero-resource language pairs. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, pages 207–215, Hanoi, Vietnam, Oct. 2020. Association for Computational Linguistics. URL <https://aclanthology.org/2020.paclic-1.24>.