Improving the Robustness to Variations of Objects and Instructions with a Neuro-Symbolic Approach for Interactive Instruction Following

Kazutoshi Shinoda[1,2] **Yuki Takezawa**[3] **Masahiro Suzuki**[1]
Yusuke Iwasawa[1] **Yutaka Matsuo**[1]
[1]The University of Tokyo [2]National Institute of Informatics [3]Kyoto University
shinoda@is.s.u-tokyo.ac.jp
yuki-takezawa@ml.ist.i.kyoto-u.ac.jp
{masa,iwasawa,matsuo}@weblab.t.u-tokyo.ac.jp

Abstract

An interactive instruction following task (Shridhar et al., 2020) has been proposed as a benchmark for learning to map natural language instructions and first-person vision into sequences of actions to interact with objects in a 3D simulated environment. We find that an existing end-to-end neural model (Shridhar et al., 2020) for this task is not robust to variations of objects and language instructions. We assume that this problem is due to the high sensitiveness of neural feature extraction to small changes in vision and language inputs. To mitigate this problem, we propose a neuro-symbolic approach that performs reasoning over high-level symbolic representations that are robust to small changes in raw inputs. Our experiments on the ALFRED dataset show that our approach significantly outperforms the existing model by 18, 52, and 73 points in the success rate on the ToggleObject, PickupObject, and SliceObject subtasks in unseen environments respectively.

1 Introduction

Instruction following, which requires an agent to understand and follow natural language instructions, has been studied to enable non-experts to operate robots (MacMahon et al., 2006). In recent years, a task called "interactive instruction following" has been proposed in order to enable agents to perform complex tasks using language instructions that require agent to interact with objects as well as to move in environments (Shridhar et al., 2020). Here, interaction with objects refers to the movement or change in the state of objects due to actions such as picking up, heating, cooling, cleaning, or cutting.

In interactive instruction following, agents need to be robust to variations of objects and language instructions that are not seen during training. For example, as shown in Figure 1, objects are of the same type but vary in attributes such as color, shape, and texture. Also, as shown in Figure 2, language instructions vary in predicates, referring expressions pointing to objects, and the presence or absence of modifiers, even though their intents are the same. However, our analysis shows that the end-to-end neural model proposed by Shridhar et al. (2020) is not robust to variations of objects and language instructions, i.e., it often fails to interact with objects with unseen attributes or to take the correct actions consistently when language instructions are replaced by their paraphrases. Similar phenomena have been observed in the existing literature. End-to-end neural models that compute outputs from vision or language inputs without any symbolic representations in the process are shown to be sensitive to small perturbations in inputs in image classification (Szegedy et al., 2013) and natural language understanding (Jia and Liang, 2017).

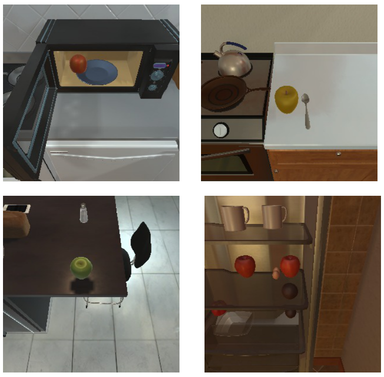
Figure 1: An example of four different apples that an agent needs to pick up, taken from ALFRED. An agent needs to interact with objects of various shapes, colors, and textures.

In this study, we aim to mitigate this problem by utilizing symbolic representations that can be extracted from raw inputs. We hypothesize that reasoning over the high-level symbolic representations of objects and language instructions are robust to variations of inputs. Specifically, high-level symbolic representations in this study refer to classes

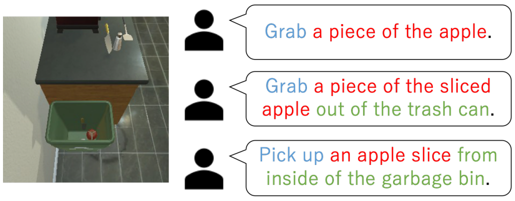
Grab a piece of the apple.
Grab a piece of the sliced apple out of the trash can.
Pick up an apple slice from inside of the garbage bin.

Figure 2: An example where different language instructions are given by different annotators to the same action, taken from ALFRED. Predicates, referring expressions, and modifiers have the same meaning but can be expressed in various ways. Modifiers can be omitted. Agents should take the correct action consistently no matter how the given instruction is expressed.

of objects, high-level actions, and their arguments of language instructions. These symbolic representations are expected to be robust to small changes in the input because of their discrete nature.

Our contributions are as follows.

- We propose Neuro-Symbolic Instruction Follower (NS-IF), which introduces object detection and semantic parsing modules to improve the robustness to variations of objects and language instructions for the interactive instruction following task.
- In subtasks requiring interaction with objects, our NS-IF significantly outperforms an existing end-to-end neural model in the success rate while improving the robustness to the variations of vision and language inputs.

2 Neuro-Symbolic Instruction Follower

We propose Neuro-Symbolic Instruction Follower (NS-IF) to improve the robustness to variations of objects and language instructions. The whole picture of the proposed method is shown in Figure 3. Each component is explained below.

2.1 Notation

The length of the sequence of actions required to accomplish a task is T. The action at time t is a_t. The observed image at time t is v_t. The total number of subtasks is N. The step-by-step language instruction for the n-th subtask is l_n, and the language instruction indicating the goal of the overall task is g. Let b_n be the high-level action for the language instruction l_n for each subtask, and r_n be its argument. The total number of observable objects in v_t is M. The mask of the m-th object is u_m, and the class of the m-th object is c_m. An example is displayed in Figure 4.

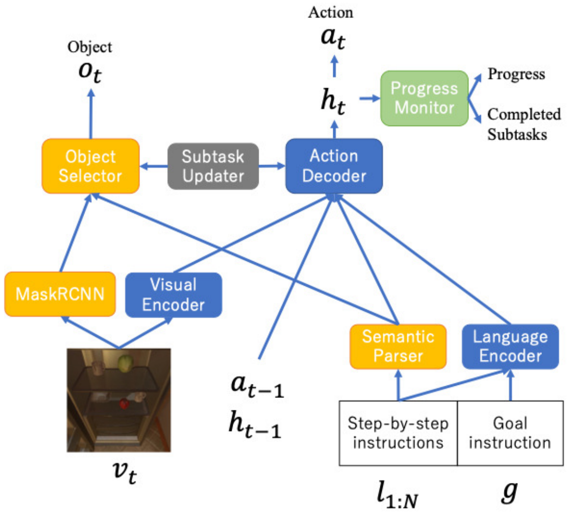

Figure 3: Overview of our model.

2.2 Language Encoder

Previous Neuro-Symbolic methods perform inference using only the symbolic representation obtained by transforming the input. However, the high-level symbolic representation of language instructions obtained in this study is only the predicate $b_{1:N}$ and the object $r_{1:N}$, and information about modifiers is lost. In order to avoid this hindrance to the success of the task, we input all the words in the language instructions to the language encoder to obtain continuous representations. The word embeddings of the language instruction g representing the goal and the step-by-step language instruction $l_{1:N}$ for all subtasks are concatenated and inputted into BiLSTM to obtain a continuous representation H of the language instruction.[1]

2.3 Visual Encoder

Similarly, for the image v_t, a continuous representation V_t is obtained with ResNet-18 (He et al., 2016), whose parameters are fixed during training.

2.4 Semantic Parser

Here, we convert the language instructions l_n for each subtask into high-level actions b_n and their arguments r_n. In this study, we used the ground truth b_n and r_n provided by ALFRED not only in training but also in testing to verify the usefulness of the symbolic representation. Predicting these labels with neural classifiers is future work.

[1] When using only high-level symbolic expressions as input to the BiLSTM, the accuracy decreased. Therefore, we use continuous representation as input here.

Goal Instruction g: Put the chilled slice apple in the trash bin

n	Step-by-step instructions l_n	High-level action b_n	Argument r_n
0	Turn right then head to the counter beside the microwave	GotoLocation	countertop
1	Pick up the knife on the counter	PickupObject	knife
2	Turn left then head to the sink	GotoLocation	apple
3	Slice the apple in the sink	SliceObject	apple
⋯	⋯	⋯	⋯
N	Put the slice apple in the trash bin	PutObject	garbagecan

(a) Instructions and their high-level actions and arguments

n	0	0	⋯	0	1	2	⋯	2	3	4	⋯
t	0	1	⋯	10	11	12	⋯	16	17	18	⋯
Action a_t	Look Down	Rotate Right	⋯	Move Ahead	Pickup Object	Rotate Left	⋯	Look Down	Slice Object	Look Down	⋯
Object o_t	-	-	⋯	-	Knife	-	⋯	-	Apple	⋯	⋯
Vision v_t			⋯				⋯				⋯

(b) Visual inputs and ground-truth actions and objects for each time step

Figure 4: An example taken from ALFRED.

2.5 MaskRCNN

MaskRCNN is used to obtain the masks $u_{1:M}$ and classes $c_{1:M}$ of each object from the image v_t. Here, we use a MaskRCNN pre-trained on ALFRED.[2]

2.6 Subtask Updater

We find that the distribution of the output action sequences varies greatly depending on which subtask is being performed. In this section, to make it easier to learn the distribution of the action sequences, we predict the subtask s_t being performed at each time. In order to verify the effectiveness of this module, we conducted an experiment under the condition that the ground truth s_t is given during both training and testing.

2.7 Action Decoder

The action decoder predicts the action a_t at each time using LSTM. The input is the hidden state vector h_{t-1} at time $t-1$, the embedding vector of the action a_{t-1}, the embedding representation of the high-level action $E(b_{1:N})^T p(s_t)$ and V_t at time t obtained using the embedding layer E and s_t, and the output x_{t-1} from h_{t-1} to H. V_t, and w_t, which is the concatenation of the output x_t of attention from h_{t-1} to H. Then, after concatenating w_t to the output h_t of LSTM, we obtain the distribution of behavior a_t via linear layer and Softmax function.

2.8 Object Selector

When the action a_t is an interaction action such as Pickup or Slice, models need to select the object with a mask. The object selector module outputs the mask of an selected object detected by MaskRCNN as follows:

$$p(o_t) = \sum_n p(s_t = n)\text{Softmax}(E(c_{1:M})E(r_n)^T) \tag{1}$$

$$m^* = \text{argmax}_{o_t} p(o_t). \tag{2}$$

Then, the model outputs the mask u_{m^*}. The overview of the object selector is shown in Figure 5.

2.9 Progress Monitor

Following Shridhar et al. (2020), our model learns the auxiliary task with the Progress Monitor, which monitors the progress of the task. Specifically, from h_t and w_t, we obtain normalized progress (t/T) and completed subtasks (number of accomplished

[2] https://github.com/alfworld/alfworld

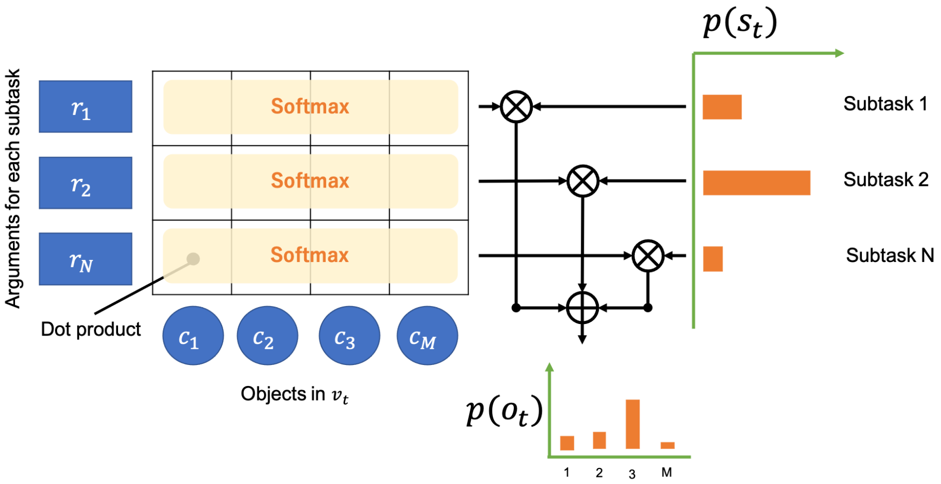

Figure 5: Detailed illustration of the object selector.

subtasks divided by N) through independent linear layers.

	Model	Goto	Pickup	Slice	Toggle
Seen	S2S+PM (Paper)	- (51)	- (32)	- (25)	- (100)
	S2S+PM (Ours)	**55** (46)	37 (32)	20 (15)	**100** (100)
	NS-IF	42 (35)	**70** (64)	**73** (59)	**100** (99)
Unseen	S2S+PM (Paper)	- (22)	- (21)	- (12)	- (32)
	S2S+PM (Ours)	26 (15)	14 (11)	3 (3)	34 (28)
	NS-IF	**28** (17)	**66** (54)	**76** (52)	**52** (52)

Table 1: Success rate (%) for each subtask. The scores that take into account the number of actions required for success are given in parentheses. Higher is better.

3 Experiments & Results

3.1 Dataset

We use the ALFRED dataset, in which roughly three annotators provided different language instructions for the final objective and each subtask for each demonstration played by skilled users of AI2-Thor (Kolve et al., 2017). ALFRED also provides the Planning Domain Definition Language (PDDL; (McDermott et al., 1998)), which contains the high-level actions and their arguments. They are used to define the subtasks when creating the dataset. In this study, we defined high-level actions and their arguments as the output of the Semantic Parser. The number of training sets is 21,023. Since the test sets are not publicly available, we use the 820 validation sets for rooms that are seen during training, and the 821 validation sets for rooms that are not seen during training. Note that the object to be selected in the validation set is an object that has never been seen during training, regardless of the room. Therefore, models need to be robust to unseen objects in both the validation sets.

3.2 Subtask Evaluation

In this study, we only evaluate the performance on each subtask, not the whole task, to verify the effectiveness of the symbolic representations. The baseline model is SEQ2SEQ+PM (Shridhar et al., 2020), which uses only continuous representations in the computation process unlike our model.

We report the results in Table 1. The proposed NS-IF model improves the success rate especially in the tasks requiring object selection, such as PickupObject, SliceObject and ToggleObject. Notably, NS-IF improved the score on SliceObject in the Unseen environments from 3% to 76% compared to S2S+PM. The fact that only objects with unseen attributes need to be selected to accomplish the tasks in the test sets indicates that the proposed method is more robust to variations of objects on these subtasks than the baseline.

On the other hand, the S2S+PM model fails in many cases and does not generalize to unknown objects. Moreover, the accuracy of S2S+PM is much lower in Unseen rooms than in Seen ones, which indicates that S2S+PM is less robust not only to unknown objects but also to the surrounding room environment. However, the difference in accuracy of NS-IF between Seen and Unseen is small, indicating that the proposed model is relatively robust to unknown rooms. This may be related to the fact that the output of ResNet is sensitive to the scenery of the room, while the output of MaskRCNN is not. The failed cases of NS-IF in PickupObject

	Model	Goto	Pickup	Slice	Toggle
Seen	S2S+PM	**315** / 240 / **239**	105 / 52 / 202	7 / 5 / 29	**29 / 0 / 0**
	NS-IF	250 / **178** / 368	**253 / 9 / 97**	**32 / 0 / 9**	**29 / 0 / 0**
Unseen	S2S+PM	147 / 99 / 513	42 / 21 / 281	1 / **0** / 31	13 / 10 / 30
	NS-IF	**165 / 89 / 502**	**218 / 12 / 113**	**25 / 0 / 7**	**28 / 0 / 25**
					(I) ↑ / (II) ↓ / (III) ↓

Table 2: Three kinds of values, (I), (II), and (III), that reflect the robustness to variations of language instructions in subtask evaluation are reported. These values represent the number of demonstrations where a model (I) succeeds with all the language instructions, (II) succeeds with at least one language instruction but fails with other paraphrased language instructions, or (III) fails with all the language instructions. Higher is better for (I), and lower is better for (II) and (III).

and SliceObject are due to the failure to predict the action a_t, or failure to find the object in drawers or refrigerators after opening them.

There are still some shortcomings in the proposed model. There was little improvement in the Goto subtask. It may be necessary to predict the bird's eye view from the first person perspective, or the destination based on the objects that are visible at each time step. In addition, the accuracy of other subtasks (PutObject, etc.) that require specifying the location of the object has not yet been improved. This is because the pre-trained MaskRCNN used in this study has not been trained to detect the location of the object.

3.3 Evaluating the Robustness to Variations of Language Instructions

The robustness of models to variations of language instructions can be evaluated by seeing whether the performance remains the same even if the given language instructions are replaced by paraphrases (e.g., Figure 2) under the same conditions of the other variables such as the room environment and the action sequence to accomplish the task.

The results are shown in Table 2. The reported values show that the proposed model increased the overall accuracy while improving the robustness to variations of language instructions compared to the baseline. The number of demonstrations corresponding to (II), "succeeds with at least one language instruction but fails with other paraphrased language instructions", was less than 4% for Pickup, 0% for Slice and 0% for Toggle, indicating that the proposed model is robust to paraphrased language instructions.

The cases that fall into the category (III), "fails with all the language instructions", are considered to be failures due to causes unrelated to the lack of the robustness to various language instructions. These failures are, for example, due to the failure to select an object in a drawer or a refrigerator after opening them.

4 Related Work

4.1 Neuro-Symbolic Method

In the visual question answering (VQA) task, Yi et al. (2018) proposed neural-symbolic VQA, where the answer is obtained by executing a set of programs obtained by semantic parsing from the question against a structural symbolic representation obtained from the image using MaskRCNN (He et al., 2017). Reasoning on a symbolic space has several advantages such as (1) allowing more complex reasoning, (2) better data and memory efficiency, and (3) more transparency, making the machine's decisions easier for humans to interpret. In the VQA task, several similar methods have been proposed. Neuro-Symbolic Concept Learner (Mao et al., 2019) uses unsupervised learning to extract the representation of each object from the image and analyze the semantics of the questions. Neural State Machine (Hudson and Manning, 2019) predicts a scene graph including not only the attributes of each object but also the relationships between objects to enable more complex reasoning on the image. However, they are different from our study in that they all deal with static images and the final output is only the answer. Neuro-Symbolic methods were also applied to the video question answering task, where a video, rather than a static image, is used as input to answer the question (Yi* et al., 2020). However, here too, the final output is only the answer to the question.

4.2 Embodied Vision-and-Language Task

Tasks that require an agent to move or perform other actions in an environment using visual and language information as input have attracted much attention in recent years. In the room-to-room dataset (Anderson et al., 2018), a Vision-and-Language Navigation task was proposed to follow language instructions to reach a destination, but it does not require interaction with objects. In both the embodied question answering (Das et al., 2018) and interactive question answering (Gordon et al., 2018) tasks, agents need to obtain information and answer questions through movement in the environment, and the success or failure is determined by only the final output answer. In contrast to these tasks, ALFRED (Shridhar et al., 2020) aims to accomplish a task that involves moving, manipulating objects, and changing states of objects in a 3D simulated environment that closely resembles reality.

5 Conclusion

In this study, we proposed a Neuro-Symbolic method to improve the robustness to variations of objects and language instructions for interactive instruction following. In addition, we introduced the Subtask Updater module that allows the model to select more appropriate actions and objects by recognizing which subtask is solved at each time step. Our experiments showed that the proposed method significantly improved the success rate in the subtask requiring object selection when the model was given the output of semantic parsing and the prior knowledge of which subtask the model was solving at each time step. The experimental results suggest that the proposed model is robust to a wide variety of objects. However, interaction with unknown objects at the class level is not required in the ALFRED evaluation dataset. Therefore, care should be taken when dealing with an unfamiliar class of objects. Furthermore, the results showed that the number of cases where a model succeeds or fails depending on the given language instructions under the same demonstration was decreased in the proposed model.

ALFRED contains the ground truth output of semantic parsing and the prior knowledge of which subtask was being solved at each step, so it was possible to use them in training and testing in this study, so it should be noted that the cost of annotations of them can not be ignored for other datasets or tasks. Additional analysis is needed to determine how much annotation is actually needed. If the cost is impractical, it may be possible to solve the problem by unsupervised learning, as in NS-CL (Mao et al., 2019). On the other hand, annotation is not necessary because the mask and class information of the object used for training MaskRCNN can be easily obtained from AI2-Thor. Therefore, whether annotation of mask and class is necessary or not depends on how well an object detection model trained on artificial data obtained from simulated environments such as AI2-Thor generalizes to real world data.

This study is still in progress. Future work includes learning of semantic parser and subtack updater to enable evaluation on the whole task.

Acknowledgements

The authors would like to thank Shuhei Kurita for his constructive feedback, and the anonymous reviewers for their valuable comments.

References

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. 2018. Embodied question answering. In *CVPR*.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. 2018. Iqa: Visual question answering in interactive environments. In *CVPR*.

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. 2017. Mask r-cnn. In *ICCV*.

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

Drew Hudson and Christopher D Manning. 2019. Learning by abstraction: The neural state machine. In *NeurIPS*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. 2017. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*.

Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. 2006. Walk the talk: Connecting language, knowledge, and action in route instructions. In *AAAI*.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *ICLR*.

Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. 1998. Pddl the planning domain definition language.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *CVPR*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Kexin Yi*, Chuang Gan*, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. 2020. Clevrer: Collision events for video representation and reasoning. In *ICLR*.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *NeurIPS*.