Contextual Interpretation in Human-Robot Dialogue

Svetlana Stoyanchev, Youmna Farag, Simon Keizer, Mohan Li, Rama Doddipatla
Cambridge Research Laboratory
Toshiba Europe

1 Introduction

The possibility of robotic assistants in homes is swiftly turning into reality as robotics companies are developing domestic robots.[1] Much like human assistants, these robotic systems offer an opportunity of continuous on-the-job learning – acquiring new skills, adapting to unfamiliar tools, and intuitively aligning with user preferences. We envision that this learning process as highly interactive and driven by natural language communication. Through task-specific instructions, real-time feedback, and iterative correction, users guide robots in mastering everyday tasks and refining their behavior.

Recent advances in large language models (LLMs) have demonstrated significant success in enabling such capabilities. LLMs can decompose high-level commands into actionable sequences (Ahn et al., 2022; Huang et al., 2022; Wang et al., 2023; Dalal et al., 2024; Shridhar et al., 2020) and even generate executable programs that map directly to robotic behaviors (Yang et al., 2024; Kranti et al., 2024).

Existing studies that leverage LLMs as language interpreters for robots have focused on 'trajectory from dialogue' task (where the model constructs a plan from provided human-human dialogue), or synchronous interactions where the user provides a command, the robot comes up with a plan of actions, executes it, and responds to the user (Shridhar et al., 2020; Padmakumar et al., 2022; Sarch et al., 2023; Zhang et al., 2022; Jiang et al., 2025). However, these paradigms assume a linear and discrete interaction flow that may not reflect real-world dynamics. We envision users engaging with robots in a more fluid and asynchronous manner by issuing new commands while the robot is still executing ongoing actions.

User Utterance	Grounded Executable Actions
Let's make coffee	Find(Mug_0);Pickup((Mug_0);# Place(Mug_0,$CoffeeMaker_0$)
This mug would not fit, use another one	Find(Mug_1);Pickup(Mug_1);# Place(Mug_1,$CoffeeMaker_0$)
Make sure to wash it first	Goto($Sink_0$);Place(Mug_1,$Sink_0$); TurnOn($Faucet_0$); TurnOff($Faucet_0$);# Pickup(Mug_1); Place(Mug_1,$CoffeeMaker_0$)
Actually bring me an apple instead	Pickup($Apple_0$);Goto($User$)

Table 1: Example of asynchronous user commands and corresponding interpretation as *Grounded Executable Actions*. # marks the timing of the next user utterance in relation to the plan execution.

As illustrated in the example in Table 1, user utterances can vary widely in granularity, contextual dependence, and timing. They may be a high-level context-free commands (e.g., *"Let's make coffee"*) or context-dependent instructions that include referring expressions distinguishing between existing and new object instances (e.g., *"Use another one"*), modifications to the current plan (e.g., *"Make sure to wash it first"*), or complete plan changes (e.g., *"Actually bring me an apple instead"*). Crucially, these utterances can be issued during the execution of an ongoing action sequence, and their timing affects their interpretation. For instance, in the human-human TEACh dataset, majority of the commander instructions occur mid-execution interrupting ongoing robot actions (Padmakumar et al., 2022), highlighting the need for systems capable of interpreting user input in the context of current actions.

While full incrementality represents the ideal paradigm for human-robot dialogue systems (Kennington et al., 2025), a minimal yet essential capability for effective interaction is the ability to process user commands issued during ongoing robot actions. In such settings, interpreting these com-

[1] https://www.1x.tech/neo, https://thehumanoid.ai/

Proceedings of the 29th Workshop on the Semantics and Pragmatics of Dialogue, September 3–5, 2025, Bielefeld, Germany.

mands requires considering both the preceding dialogue context and the status of the current execution plan. In this paper, we describe a study aiming to evaluate LLMs – now widely adopted as general-purpose language processing tools – as interpreters of natural language commands for a robot assistant. The evaluation includes scenarios where commands are issued asynchronously while the robot is actively performing tasks.

2 Method

2.1 System

To collect realistic interaction examples, we develop a multimodal text and video interface, integrated with AI2THOR simulator, featuring kitchens, living room, and bathroom environments (Kolve et al., 2017; Padmakumar et al., 2022). Users interact with the virtual robot through text-based chat, observing the environment and robot actions through ego-centric view of the robot's camera. To support asynchronous interaction, the interface pauses before executing each action in the plan, giving the user an option to interject with new input. We use an LLM to convert user's natural language (NL) utterance, contextual information—such as the current plan and previously referenced objects, into a sequence of executable API calls for the robot actions (see Table 1).

The interface is designed to support two distinct user roles: (1) a naive end-user who interacts with the robot using natural language, and (2) an expert user who provides natural language input along with corrections of the interpretation when necessary. To ensure that the experts are equipped to accurately correct the system interpretations, they are first required to complete five tasks by directing the robot using grounded executable commands.

2.2 Experiment design

The objective of this study is to collect realistic human–robot interaction data, including natural language utterances issued while the robot is actively performing actions. Four expert users are each instructed to complete tasks, including making coffee, collecting objects, and cleaning dishes, across ten distinct home environments by providing (1) step-by-step instructions and (2) high-level directives. Users are encouraged to interrupt the system whenever necessary to refine or redirect its behavior.

Each NL input is interpreted by GPT-4.1, and the resulting interpretation is displayed to the expert user, who may choose to: (a) continue the interaction using the automatic interpretation, (b) correct the interpretation by expressing their intent as a sequence of executable commands, or (c) issue a new command, disregarding the previous utterance and proceeding with the interaction.

2.3 Preliminary statistics

Table 2 shows the preliminary statistics from the ongoing data collection. From the total of 267 expert user utterances collected so far, the interpretation of 20.6% was manually corrected. 176 were non-interrupting utterances, including the initial utterances of a dialogue and those following a successful execution of the previous plan. These were manually corrected in 16% of cases. The remaining 91 utterances interrupted the robot's actions. Among these, we distinguish between the utterances following a robot failing to execute an action (FAIL EXEC) and those following a successful action execution (SUCCESS EXEC). Manual corrections to the automatically generated plan were applied in 12.5% of FAIL EXEC and 36.5% of SUCCESS EXEC interrupting utterances.

Type	**# Utterances**	**% Corrected**
Total	267	20.6%
Initial/no interrupt	176	16%
After FAIL EXEC	38	12.5%
After SUCCESS EXEC	53	36.5%

Table 2: Statistics on the utterance types and the rate of manual correction of interpretation by experts.

3 Summary and Future Work

In this work, we introduced a multimodal virtual robot interface that enables users to interrupt the system while it is executing actions. We proposed a novel data collection methodology in which expert users serve as both participants and annotators, streamlining the annotation process by embedding it directly into the interaction.

Preliminary analysis reveals that while the overall misinterpretation rate by GPT-4.1 is 20%, in-context interruptions are misinterpreted significantly more often, at a rate of 36.5%. In future work, we will leverage the complete dataset to conduct experiments aimed at improving interpretation accuracy, particularly for in-context utterances. Planned directions include dynamic selection of few-shot examples from the repository of expert annotations and the incorporation of chain-of-thought reasoning into these examples.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*.

Murtaza Dalal, Tarun Chiruvolu, Devendra Chaplot, and Ruslan Salakhutdinov. 2024. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. In *The Twelvth International Conference on Learning Representations ICLR 2024*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR.

Chenxi Jiang, Chuhao Zhou, and Jianfei Yang. 2025. REI-bench: Can embodied agents understand vague human instructions in task planning? *Preprint*, arXiv:2505.10872.

Casey Kennington, Pierre Lison, and David Schlangen. 2025. Prior lessons of incremental dialogue and robot action management for the age of language models. *Preprint*, arXiv:2501.00953.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. 2017. AI2-THOR: An Interactive 3D Environment for Visual AI. *Preprint*, arXiv:1712.05474.

Chalamalasetti Kranti, Sherzod Hakimov, and David Schlangen. 2024. Towards no-code programming of cobots: Experiments with code synthesis by large code models for conversational programming. *Preprint*, arXiv:2409.11041.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2022. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025.

Gabriel Sarch, Yue Wu, Michael Tarr, and Katerina Fragkiadaki. 2023. Open-ended instructable embodied agents with memory-augmented large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian (Shawn) Ma, and Yitao Liang. 2023. Describe, explain, plan and select: Interactive planning with llms enables open-world multi-task agents. In *Advances in Neural Information Processing Systems*, volume 36, pages 34153–34189.

Jingkang Yang, Yuhao Dong, Shuai Liu, Bo Li, Ziyue Wang, Chencheng Jiang, Haoran Tan, Jiamu Kang, Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. 2024. Octopus: Embodied vision-language programmer from environmental feedback. In *The 18th European Conference on Computer Vision ECCV 2024*.

Yichi Zhang, Jianing Yang, Jiayi Pan, Shane Storks, Nikhil Devraj, Ziqiao Ma, Keunwoo Yu, Yuwei Bao, and Joyce Chai. 2022. DANLI: Deliberative agent for following natural language instructions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1280–1298, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.