Enhancing Human-Robot Interaction with IDRA-H

Kira Herlemann (Mat.Nr. 12229970), Matthias Hirschmanner, Peter Hönig and Markus Vincze

***Abstract*—Service robots need to be controlled by users without programming knowledge. Natural language is one possible modality and inherently intuitive. This paper introduces Intelligent Dialogue for Robot Applications with Humans (IDRA-H) to enable intuitive user interaction with a robotic grasping pipeline, specifically simulated on the Toyota Human Support Robot (HSR). The proposed system uses Large Language Models (LLMs) to extract information from user dialogues to control a robot for pick-and-place actions. The approach incorporates prompting the LLM with a system message and dialogue examples to enhance the LLM's reasoning abilities. We conduct evaluations using four models, GPT-3.5 Turbo, GPT-4o, Llama 3, and Mixtral, across various task scenarios, demonstrating their ability for reproducible behaviour. Our results highlight the influence of phrasing of the request on the consistency of LLMs responses. The proposed system is available at: https://gitlab.tuwien.ac.at/e376-acin/v4r/education/robot-vision/2024/chat-hsr.**

I. Introduction

Conversation is the most intuitive way for humans to communicate with each other. The increasing importance of Human-Robot Interaction (HRI) highlights the need to make this interaction as natural as possible. LLMs enable promising competencies in processing natural language prompts, ensuring accurate responses. The existing system is integrated on the HSR and can complete pick-and-place tasks with objects on a table. As of now, there is no interaction with a user. The robot is operated by a grasping pipeline programmed as a state machine that calls upon different functions. Once the objects on the table have been identified, the system selects the nearest object. In advance, the decision is taken about whether the object will be handed over or placed on the shelf.

We propose integrating a LLM into the system and, therefore, introduce IDRA-H. This enables flexibility and particularly intuitiveness concerning the operative user. The system allows for the selection of objects and tasks based on individual requirements. The function of the LLM is to analyze the inquiry and provide the requested object and task to the state machine. The new system is evaluated with four different LLMs: GPT-3.5 Turbo and GPT-4o [1] and the open-source local models Llama 3 and Mixtral [2].

II. Related Work

Involving human requests in robotic applications introduces the need to translate natural language. Huang et al. show the possibility of using LLMs‘s reasoning capabilities for HRI. They find that a feedback loop between the LLM and the user or scene enhances performance [3]. Additionally, they demonstrate that LLMs can extract realizable details from instructions without further training when prompted suitable [4]. Introducing feedback and a precise prompt encounters the challenge of consistent responses among different requests [5].

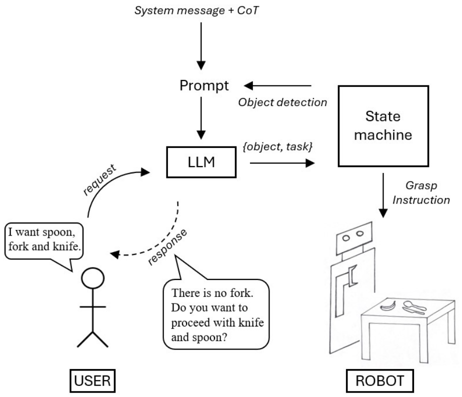


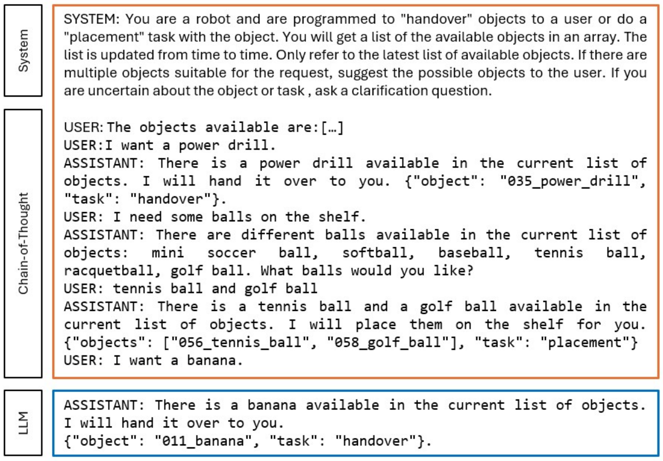
Fig. 1. IDRA-H: The LLM processes a user request on the loop (prompted with system message, chain of thoughts and the available objects). Determined object and task are provided to the state machine to instruct the robot.

The behaviour of the LLM can be shaped through prompting. With a chat between the LLM and different perception modules, until sufficient information is gained, a user request can be executed [6]. LLMs can provide a planning goal derived from natural language instructions. However, they are not used for the actual planning process since they are primarily trained for natural language processing [7]. Even though ProgPrompt is an approach to prompting LLMs to generate complete code and function calls for task planning. This highlights the importance of presenting general information about the situation to receive executable instructions from the LLM [8].

III. IDRA-H

We aim to enable the interaction between robots and humans through natural dialogue. With IDRA-H, we introduce a system that incorporates LLMs to extract information from user requests and get accurate responses regarding both style and content. The system consists of three principal components. The human requests objects, the robot grasps the objects from a table, and the LLM links the two. The robot is integrated into the system by a state machine that calls the functions for object detection and the final instruction to execute the respective grasp and task.

Fig. 1 illustrates the new system procedure. The LLM is prompted with a system message and a chain of thoughts to improve reproducibility (e.g. [9]). Furthermore, the prompt contains a list of the available objects developed by the state machine. The user requests an object. If the request includes all the needed details for the object and task, the LLM provides them to the state machine. If further information is required, it will result in a natural dialogue between the user and


System
Chain-of-Thought
LLM

SYSTEM: You are a robot and are programmed to "handover" objects to a user or do a "placement" task with the object. You will get a list of the available objects in an array. The list is updated from time to time. Only refer to the latest list of available objects. If there are multiple objects suitable for the request, suggest the possible objects to the user. If you are uncertain about the object or task , ask a clarification question.

```
USER: The objects available are:[…]
USER:I want a power drill.
ASSISTANT: There is a power drill available in the current list of objects. I will hand it over to you. {"object": "035_power_drill", "task": "handover"}.
USER: I need some balls on the shelf.
ASSISTANT: There are different balls available in the current list of objects: mini soccer ball, softball, baseball, tennis ball, racquetball, golf ball. What balls would you like?
USER: tennis ball and golf ball
ASSISTANT: There is a tennis ball and a golf ball available in the current list of objects. I will place them on the shelf for you. {"objects": ["056_tennis_ball", "058_golf_ball"], "task": "placement"}
USER: I want a banana.
```

```
ASSISTANT: There is a banana available in the current list of objects.
I will hand it over to you.
{"object": "011_banana", "task": "handover"}.
```

Fig. 2. Prompting system message and chain of thoughts followed by a user request and the model output.

LLM. The accepted response format is a JSON-style object that is easy to process by the state machine, e.g., {“object”: ‘011_banana’, “task”: ‘handover’}. The subsequent steps for finding an appropriate grasp and instructing the robot stay the same. An updated list of available objects is sent to the LLM after each finished grasping instruction. Each following interaction is based on the newest list of available objects.

For this work, the LLM endpoint is called through its REST API. OpenAI [10] provides Python libraries to interact with their API online. The API provided by Ollama [11] runs the other two models locally. Every model’s temperature attribute is set to zero for consistent outputs between trials. In addition to that, we prompt the models with a short and precise system message, where we introduce general information. Further, the prompt contains a chain of thoughts (e.g. [7]) with a list of the available objects and dialogue examples. The thoughts are decomposed into an object availability check, determining if handover or placement, and the valid JSON-style object. The full prompt can be seen in Fig. 2.

IV. Evaluation & Results

IDRA-H is evaluated on four different LLMs. Recent research suggests that these models may be suitable for our purpose [5], [12]. Llama 3 and Mixtral are chosen because of their open-source local availability. The models of OpenAI are robust and inherently intuitive [5]. The evaluation is carried out on three different user requests. The first is repeatedly asking for the same banana, which should first lead to a handover and then to a clarification. The second is preparing a fruit salad, for which the available fruits should be handed over after at most one clarification question. The third is for setting the table with a spoon, knife, fork, and plate, whereby no fork is available on the table. This should lead to the handover of the three available objects only after a one-time clarification about the fork. In addition, the final responses need to contain the correct JSON-style object.

TABLE I
Results from Experiment 1 for the 4 models on task 1-3.

Experiment 1			
LLM	1st	2nd	3rd
GPT-3.5 Turbo	10	10	9
GPT-4o	10	10	10
Llama 3	10	9	8
Mixtral	9	10	9

First, we evaluate the system ten times with the exact same queries to assess their determinism, ensuring that identical inputs yield identical outputs in terms of content. Table 2 illustrates the number of successful outcomes for each of the requests. GPT-4o is deterministic in all attempts. GPT-3.5 Turbo once did not hand over the plate in task 3. Llama 3 once handed over a marker instead of a fruit in task 2 and twice did not ask about the fork before handing over the rest in task 3. Mixtral is missing the knife twice. Additionally, when an object is unavailable (e.g. the banana in task 1 and the fork in task 3), Mixtral provides a JSON-style with the object ‘none’. The latter occurs in the second evaluation as well. It may be possible to eliminate this if such an example was included in the prompt.

Second, we test the system with three random users given instructions for the three requests without the exact phrasing. This estimates how the models perform when the requests are semantically identical but not verbatim. Except for Llama 3, the users felt comparable comfort with the models. This is because the responses of Llama 3 are more extended, as they include more non-required information. All four models reach the main task at the end. Even if they can’t produce consistent outputs on the varying user inputs, they only need one to three more dialogue shots between the model and the user giving feedback. This qualifies all of them for their use with IDRA-H.

V. Conclusion

This work introduces IDRA-H, a system that intuitively controls a robot via natural language to conduct object handover and placement tasks. Getting reproducible request analysis and receiving desired response phrasing with all four models is possible. Their performance depends on the phrasing of the user request. Enhancing performance by engineering the prompt to an individual LLM rather than using the same prompt across different models is promising. In the future, we can include feedback from the state machine, such as the success of the manipulation task. It is desirable to prompt the LLM with constraints of the robot, e.g., accessibility of the objects regarding size, shape, location, or structure.

References

[1] OpenAI, “Openai documentation,” 2024. [Online]. Available: https://platform.openai.com/docs/overview

[2] Ollama, “Models,” 2024. [Online]. Available: https://ollama.com/library

[3] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, P. Sermanet, N. Brown, T. Jackson, L. Luu, S. Levine, K. Hausman, and B. Ichter, “Inner monologue: Embodied reasoning through planning with language models.” arXiv:2207.05608v1 [cs.RO], 2022.

[4] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," pp. 9118–9147, 2022.

[5] S. H. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," *IEEE Access*, vol. 12, pp. 55 682–55 696, 2024.

[6] X. Zhao, M. Li, C. Weber, M. B. Hafez, and S. Wermter, "Chat with the environment: Interactive multimodal perception using large language models," pp. 3590–3596, 2023.

[7] Y. Xie, C. Yu, T. Zhu, J. Bai, Z. Gong, and H. Soh, "Translating natural language to planning goals with large-language models." arXiv 2302.05128 [cs.CL], 2023.

[8] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 11 523–11 530.

[9] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 24 824–24 837.

[10] OpenAI, "Openai python api library," 2024. [Online]. Available: https://github.com/openai/openai-python

[11] Ollama Team, "Ollama," 2024. [Online]. Available: https://github.com/ollama/ollama

[12] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. B. Hanna, F. Bressand, and et al., "Mixtral of experts." arXiv preprint arXiv:2401.04088, 2024.