2024 IEEE 20th International Conference on Automation
Science and Engineering (CASE)
August 28 - September 1, 2024. Bari, Italy

2024 IEEE 20th International Conference on Automation Science and Engineering (CASE) | 979-8-3503-5851-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/CASE59546.2024.10711671

Task Planning for Robot Manipulator Using Natural Language Task Input with Large Language Models

Tomoya Kawabe, Tatsushi Nishi, Ziang Liu, Tomofumi Fujiwara

Abstract— **The utilization of robots is expanding beyond the industrial sector and reaching into society. This paper considers the implementation of robots in retail stores for stocking products on shelves. To optimize the task input for users unfamiliar with robots minimizing the robot's operational time is essential. This paper presents a novel task planning method using natural language input. Our proposed method converts natural language tasks into symbolic sequence representations using Large Language Models (LLM) and then the optimal task procedures are derived by executing task planning based on Monte Carlo Tree Search (MCTS). To improve the accuracy of the conversion, we propose an interactive method that allows users to confirm the conversion results to improve the correctness of the response of LLM. We incorporate a mechanism for handling error messages caused by unexpected outputs from LLM. A few-shot prompting method is adopted to guide the LLM to higher performance. Computational experiments demonstrate that the proposed method can successfully identify the intended operational procedures for approximately 90% of 18 natural language tasks.**

I. INTRODUCTION

The utilization of robots is expanding beyond industrial applications to encompass society as a whole. Robot manipulators are introduced to replace human tasks into automated replenishment, stocking, and retrieval of several types of items in retail stores. This paper presents the implementation of task and motion planning for robots in retail stores, specifically for the task of stocking products on shelves. Two aspects need to be addressed to accomplish this objective. Firstly, it is necessary to let users unfamiliar with robots input tasks in a manner that is easy to understand. Secondly, there is a need to find an optimal task procedure to minimize the robot's operational time.

Task planning has seen the proposal of a variety of conventional methods, including PDDL [1] [2] and STRIPS [3]. In this paper, to optimize task procedures to minimize the robot's operation time, we use a Monte Carlo Tree Search (MCTS)-based Task and Motion Planning (TAMP) approach [4]. This approach enables simultaneous optimization of the robot's motion trajectory, task procedures, and product placement to minimize operation time using a unique symbolic sequence task representation. However, unfamiliar users for the task planning are requested to input symbolic task sequences. As the complexity of problems increases, the number of symbolic representations also grows, making it difficult for users to memorize and use them all.

To address this challenge and facilitate easy task input for users unfamiliar with robots, we utilize Large Language Models (LLM), a recent focus of attention that includes the GPT series [5] [6] and BART [7]. LLMs trained on vast text data exhibit high accuracy in tasks such as text generation and summarization. By employing LLM, we convert natural language tasks provided by humans into symbolic sequence representations interpretable by robots. Several task planning methods leveraging LLM have already been proposed, with SayCan [8] being a representative example. SayCan [8] utilizes LLM to achieve robot action prediction. Additionally, Huang et al. [9]. and Ramon et al. [10]. have used LLM for natural language to symbolic sequence representation conversion. Jiang et al. [11] proposed LLM-driven language model programs for embodied control.

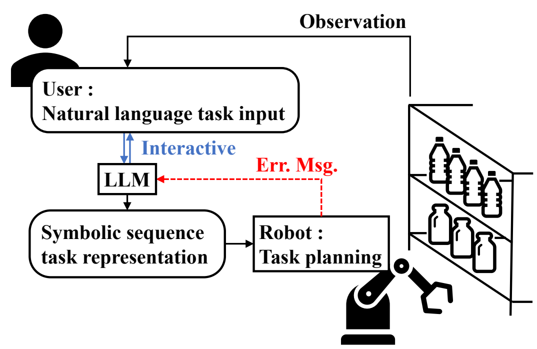


Fig. 1. Overview of the method proposed in this paper.

The overview of the proposed method is presented in Fig. 1. The proposed method consists of four main components: natural language task input, LLM, symbolic sequence representation, and task planning. The user inputs natural language tasks, which are then converted into symbolic sequence representations using LLM. Task planning is then executed based on the symbolic sequence representation. To prevent unintentional conversion of user-intended natural language tasks into unintended symbolic sequences, we propose an interactive method in which users can confirm the conversion results and provide feedback. Furthermore, to address occasional unexpected outputs from LLM, such as the use of undefined symbolic representations or clearly infeasible tasks, an error message feedback mechanism is also incorporated.

This research is subsidized by New Energy and Industrial Technology Development Organization (NEDO) under a project JPNP20016. This paper is one of the achievements of joint research with and is jointly owned copyrighted material of ROBOT industrial Basic Technology Collaborative Innovation Partnership.

Tomoya Kawabe, Ziang Liu, Tatsushi Nishi, and Tomofumi Fujiwara are with Graduate School of Natural Science and Technology, Okayama University, 3-1-1 Tsushima-Naka, Kita-ku, Okayama 700-8530, JAPAN nishi.tatsushi@okayama-u.ac.jp

The contributions of this paper are as follows:

- We propose a task planning method for robot manipulators using natural language task input. Through prompt engineering, we achieve a high success rate in converting natural language tasks into symbolic sequences as the user's intention.
- We develop an interactive method that allows users to confirm the conversion results and provide feedback, as well as a mechanism for handling error messages caused by unexpected outputs from LLM.
- The LLM is integrated into the task planning process, enabling the robot to understand natural language tasks and execute task planning based on the symbolic sequence representation.

II. PROBLEM DESCRIPTION

A. Task Planning for Stocking Shelves in Retail Store

We consider the task of stocking products on shelves in retail stores. The task involves moving products from outside the shelf to inside the shelf, aligning products inside the shelf, and moving products from inside the shelf to outside the shelf. The goal is to minimize the robot's operational time by optimizing the task procedures. Fig. 2 illustrates the environment of problem setting. The total system consists of a single robot manipulator, one shelf, and multiple objects.

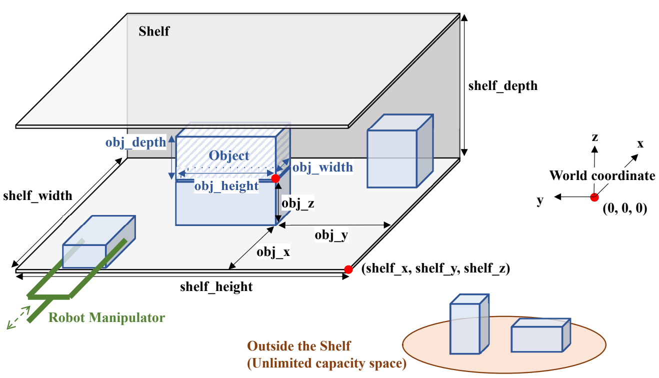


Fig. 2. Problem environment.

The shelf is composed of a single tier, with capacity constraints within the shelf and unlimited capacity outside the shelf. Objects can be placed inside or outside the shelf and are represented as rectangular prisms with unique sizes. The robot manipulator executes operations by switching between four types of grippers: long and wide, long and narrow, short and wide, and short and narrow. The narrow gripper can grasp objects with widths ranging from 0 to L [m], while the wide gripper can grasp objects with widths ranging from L to $L_{\max}$ [m], where $L_{\max}$ is the maximum width of the object. Additionally, the long-clawed gripper can operate on objects regardless of their position within the shelf but cannot lift them. On the other gripper, the short-clawed gripper can only operate on objects placed in front of the shelf. However, it is capable of lifting them.

B. Definition of Operations for Robot Manipulator

The feasible operations for the robot are the sequence of robot actions defined by "Change", "Forward", "Slide", "Jump", "In", and "Out", as outlined in Table I, which presents the executable conditions and the corresponding operation times for each action. As shown in the table, we assume the operation times for actions "Forward" and "Slide" are the sum of a constant time T_d and the product of the object's movement distance $|v|$ and a coefficient R_d. The operation times for action "change", "Jump", "In", and "Out" are the constant times T_c, T_{p3}, T_{p1}, and T_{p2}, respectively. The robot manipulator can select one of these actions at each time step to execute the overall task. The detailed definitions of each action are as follows:

TABLE I

DEFINITION OF ACTIONS THAT CAN BE SELECTED BY THE ROBOT.

Action	Argument(s)	Prerequisite(s)	Operation time [sec]
Change	$gripper_{type}$	-	T_c
Forward	k, obj_x^k, obj_y^k	$obj_x^k \geq 0 \wedge obj_z^k = 0 \wedge \mathrm{LH} \wedge \mathrm{C1}$	$T_d + R_d\,\lvert v\rvert$
Slide	k, obj_y^k	$obj_x^k = 0 \wedge obj_z^k = 0 \wedge \mathrm{LH} \wedge \mathrm{C1}$	$T_d + R_d\,\lvert v\rvert$
Jump	k, obj_y^k, obj_z^k	$obj_x^k = 0 \wedge \mathrm{SH} \wedge \mathrm{C2}$	T_{p3}
In	k, obj_y^k, obj_z^k	$obj_x^k < 0 \wedge \mathrm{SH} \wedge \mathrm{C2}$	T_{p1}
Out	k	$obj_x^k = 0 \wedge \mathrm{SH}$	T_{p2}

LH : Robot manipulator requires a long length.
SH : Robot manipulator requires a short length.
C1 : No collision with other objects while moving.
C2 : No collision of the destination with other objects.

Change: This action can change the gripper type of the robot manipulator using a gripper changer. The argument is the type of the new manipulator, and there are no specific prerequisites. The operation time is a constant time T_c.

Forward: This action can slide the object to any position on the shelf. The arguments are the object number k to be moved and the destination coordinates (obj_x, obj_y) of the object k. Feasibility conditions include the object k being located inside the shelf, the object's z-coordinate being 0, and the manipulator being of long length. The operation time is the sum of a constant time Td and the product of the object's movement distance $|v|$ and a coefficient Rd.

Slide: This action can slide the object parallel to the y-axis in front of the shelf. The arguments are the object number k to be moved and the destination y-coordinate obj_y of the object k. Feasibility conditions include the object being located in front of the shelf and meeting the feasibility conditions of "Forward". The operation time is the same as that of "Forward".

Jump: This action can move the object from the front of the shelf to any position within the shelf, restricting the x-coordinate to 0. The arguments are the object number k to be moved and the destination coordinates (obj_y, obj_z) of the object k. Feasibility conditions include the object being located in front of the shelf, the manipulator being of short length, and no collision with other objects at the destination. The operation time for the jump operation is a constant time T_{p3}.

In: This action can bring the object from outside the shelf to inside the shelf. The arguments are the same as "Jump". Feasibility conditions include the object being located outside the shelf, and the other conditions are the same as "Jump". The operation time is a constant time T_{p1}.

Out: This action can transport the object from inside the shelf to outside the shelf. The argument is the object number k to be targeted. Prerequisites are the same as "Jump", excluding collision avoidance at the destination. The operation time is a constant time T_{p2}.

C. Symbolic Sequence Representation Tasks

Users input tasks using natural language expressions such as "Move all objects outside the shelf" or "Align all objects inside the shelf". However, these natural language expressions can not be interpreted directly by the robot's task planner. In our previous work [4], we have proposed a task planning method using symbolic sequence representations. In the current work, we use the LLM to convert natural language tasks into symbolic sequence representations, which are then used as inputs for the task planner. There are four types of symbolic sequence representations that can be used for the task planner. Those representations are defined in Table II.

TABLE II
DEFINITION OF SYMBOLIC SEQUENCE REPRESENTATION FOR TASK INPUT.

Symbolic sequence	Format Type	Format Examples	Intention
out	bool	obj^1.out	Move obj^1 outside the shelf.
alignment	bool	obj^1.alignment	Align obj^1 inside the shelf.
out_limit	bool	obj^1.out_limit	Restrict operations on obj^1 to inside the shelf only.
move_limit	bool	obj^1.move_limit	Do not allow operations on obj^1.

The data types of the symbolic sequence representations are boolean. The users' intentions are also provided in the table. As shown in the table, the symbolic sequence representation "out" is used to move objects outside the shelf, "alignment" is used to align objects inside the shelf, "out_limit" is used to restrict operations on objects to inside the shelf only, and "move_limit" is used to prevent operations on objects.

Here are some examples of symbolic task inputs:

- (obj^1.out=True) (obj^2.out=True) (obj^3.out=True)
  // Move obj^1 – obj^3 outside the shelf.
- (obj^1.alignment=True) (obj^2.alignment=True) (obj^3.alignment=True)
  // Align obj^1 – obj^3 inside the shelf.
- (obj^1.out=True) (obj^2.out=True) (obj^3.move_limit=True)
  // Move obj^1 and obj^2 outside the shelf, but do not allow operations on obj^3.
- ((obj^1.alignment=True) (obj^2.alignment=True) (obj^1.out_limit=True)
  // Align obj^1 and obj^2 inside the shelf, and restrict operations on obj^1 to inside the shelf only.

Although the symbolic sequence representations are easy to understand, it is still challenging for users to memorize and use them all. In addition, inputting tasks using symbolic sequence representations is not intuitive for users unfamiliar with programming. Therefore, we propose a method that allows users to input tasks using natural language expressions, which are then converted into symbolic sequence representations using the LLM.

III. OUR PROPOSED METHOD

A. Outline of the Proposed Algorithm

The flowchart for executing task planning from natural language tasks using LLM is shown in Fig. 3. The algorithm can be divided into two parts: the portion that converts user-input natural language tasks into symbolic sequence representations and the part that executes task planning based on the symbolic representations.

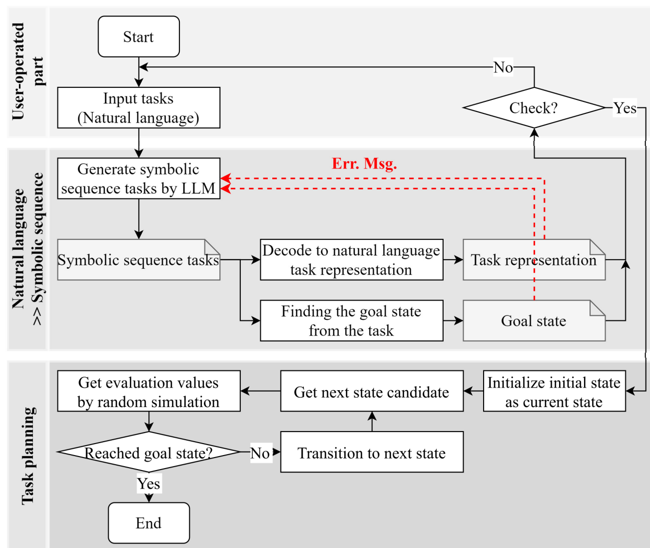


Fig. 3. Flowchart for executing task planning from natural language tasks. The algorithm incorporates a loop structure to achieve the transformation as intended by the user. Additionally, to ensure it does not terminate even if unexpected conversions occur, a process has been added to input error messages to the LLM.

As shown in the figure, first, the user inputs a task in natural language. Then, the task is converted into a symbolic sequence representation using LLM. To ensure the conversion is successful, we propose an interactive method that allows users to confirm the conversion results and provide feedback. If the conversion is unsuccessful, the user can correct the task by inputting comments in natural language. The process is repeated until the conversion is successful. In addition, an error message feedback mechanism is incorporated to handle unexpected outputs from LLM. If the conversion is successful, the task planning process is executed based on the symbolic sequence representation. This study uses MCTS as the task planning method.

In the following sections, we describe the details of the algorithm, including the conversion from natural language to symbolic sequence representations and the task planning process.

B. Conversion from Natural Language to Symbolic Sequence

In our proposed method, when the user inputs a task in natural language, it is first converted into a symbolic sequence task representation by a Large Language Model (LLM). From the output symbolic sequence task representation, a task in natural language and the target state are further generated, enabling the user to confirm whether the intended transformation has been achieved. If the transformation is successful, the process proceeds to the planning phase. Otherwise, the user can correct the task by inputting comments in natural language, and the process is repeated until the desired transformation is achieved.

In addition, an error message feedback mechanism is incorporated to handle unexpected outputs from LLM. This could happen if the LLM conversion is not successful or if undefined symbolic representations are used. In such cases, the user can input an error message to the LLM, prompting a redo of the conversion from natural language to symbolic sequence. Once the user confirms the desired transformation, the process proceeds to the planning phase. In this study, we used GPT-3.5 [6] as the LLM.

Many previous studies have suggested that conducting prompt engineering is important for improving the performance of LLMs. This study also involves prompt engineering to achieve the desired transformation. In detail, we use a few-shot prompting method, which involves providing some demonstrations within the prompt to guide the model to higher performance. The example prompt for the case of 5 objects is shown in Table III.

C. Task Planning

Task planning is executed using our original Monte Carlo Tree Search (MCTS). After the user confirms the symbolic sequence representation, the generated symbolic sequence representation is used as the input for the task planner. Then, the task planner explores the state transitions from the initial state to the goal state using MCTS. The process of task planning is described in the following steps.

Step 1. Define the initial state n_s as n_t.

Step 2. Enumerate candidate next states $\{n_{t+1}\}$ that can be transitioned from the current state n_t.

Step 3. Explore state transitions from the enumerated candidate next states $\{n_{t+1}\}$ through random search until reaching the goal state n_g.

Step 4. Upon reaching the goal state n_g, estimate the operation time from the initial state n_s to the goal state n_g, considering it as the evaluation value for the candidate's next state n_{t+1}.

Step 5. Transition to the candidate's next state n^*_{t+1} with a high evaluation value, making it the current state n_t.

Step 6. Repeat steps 2 to 5 until the current state n_t matches the goal state n_g.

The evaluation value of each node, as shown in Fig. 4, is represented by the sum of the known operation time from the initial state to the candidate's next state node and the estimated operation time from the candidate's next state node to the goal state. The estimated operation time to the goal state is determined by the average value obtained from 50 repetitions of random simulations. A detailed explanation of the task planning method is provided in our previous work [4].

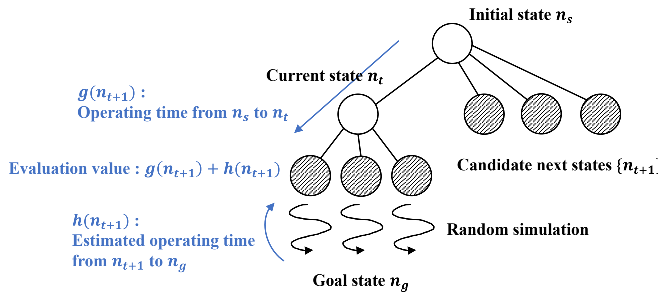


Fig. 4. Definition of the evaluation value of the next state node.

IV. COMPUTATIONAL EXPERIMENTS

A. Case Study

Fig. 5 illustrates an example of the execution of the proposed method. Ten objects are placed inside the shelf at the beginning, as shown in Fig. 5 (a). Then, the user inputs the task "Retrieve all items from inside the shelf" in natural language. The task is then converted into a symbolic sequence representation using LLM. The symbolic sequence representation is then used as the input for the task planner, which explores the state transitions from the initial state to the goal state using MCTS. The sequence of actions obtained is shown in the figure. The robot manipulator successfully moved all objects outside the shelf as instructed. This example demonstrates the successful execution of the proposed method.

B. Comparison with Other Methods

To verify the effectiveness of the proposed method, we conduct the computational experiments to compare it with other conventional methods. The compared methods are explained as follows:

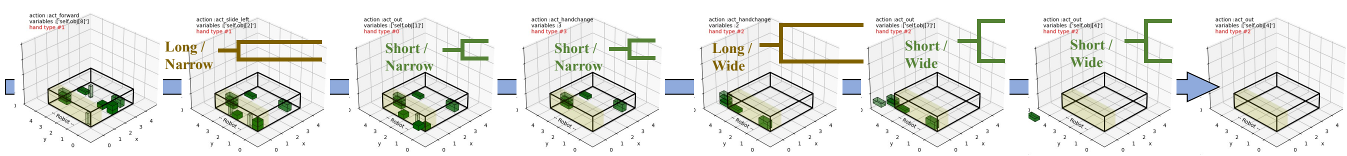
- **Non-Interactive**: A method that does not involve interactive communication between the user and the robot or feedback of error messages.
- **Interactive**: A method that involves interactive communication between the user and the robot but does not include feedback on error messages.
- **Interactive (err. FB)**: The proposed method involves interactive communication between the user and the robot and includes feedback on error messages.

The extraction problem and arrangement problem were used as the natural language tasks. The extraction problem involves moving objects outside the shelf, while the arrangement problem involves aligning objects inside the shelf. For

TABLE III

EXAMPLE PROMPT FOR THE CASE OF 5 OBJECTS.

Please portray yourself as a robot. The user will assign tasks related to five objects: obj1, obj2, obj3, obj4, and obj5. The robot should convert these tasks into symbolic representations and output them. Obj1 belongs to product group A, obj2, obj4, and obj5 belong to product group B, and obj3 belongs to product group C. Obj1, obj2, and obj3 are currently inside the shelf, while obj4 and obj5 are currently outside the shelf. user: I want to move obj1, obj2, and obj4 outside the shelf. robot: obj1.out = True, obj2.out = True, obj4.out = True user: I want to move all objects outside the shelf. robot: obj1.out = True, obj2.out = True, obj3.out = True, obj4.out = True, obj5.out = True user: Do not allow operations on obj1 when moving obj2 and obj5 outside. robot: obj2.out = True, obj5.out = True, obj1.move_limit = True user: I want to prevent obj3 from being taken outside when moving obj3 to obj5. robot: obj3.out = True, obj4.out = True, obj5.out = True, obj3.out_limit = True user: I want to arrange obj2, obj4, and obj5 inside the shelf. robot: obj2.alignment = True, obj4.alignment = True, obj5.alignment = True user: I want to arrange all objects inside the shelf. robot: obj1.alignment = True, obj2.alignment = True, obj3.alignment = True, obj4.alignment = True, obj5.alignment = True user: Do not allow operations on obj1 and obj5 when arranging obj3 and obj4. robot: obj3.alignment = True, obj4.alignment = True, obj1.move_limit = True, obj5.move_limit = True user: Limit operations to inside the shelf when arranging obj1 to obj4, except for obj2 and obj3.robot: obj1.alignment = True, obj2.alignment = True, obj3.alignment = True, obj4.alignment = True, obj2.out_limit = True, obj3.out_limit = True user: I want to arrange obj1 to obj3 inside the shelf. robot: obj1.alignment = True, obj2.alignment = True, obj3.alignment = True user: I want to keep obj1 inside the shelf. robot: obj1.alignment = True, obj2.alignment = True, obj3.alignment = True, obj1.out_limit = True user: I want to take out obj2 and obj4 outside the shelf. robot: obj2.out = True, obj4.out = True user: Please do not move obj3 and obj5. robot: obj2.out = True, obj4.out = True, obj3.move_limit = True, obj5.move_limit = True robot: obj1.out = True, obj2.out = True, obj5.alignment = True user: Do not use “out” and “alignmen” simultaneously. robot: obj5.alignment = True robot: obj2.out = True, obj3.alignment = True, obj2.alignment = True, obj2.out_limit = True user: Do not use “out” and “alignmen” simultaneously. robot: obj2.alignment = True, obj3.alignment = True, obj2.out_limit = True robot: obj1.out = True, obj3.alignment = True, obj1.out_limit = True user: Do not use “out_limit” and “out” simultaneously for obj1. robot: obj1.out = True, obj2.alignment = True robot: obj1.alignment = True, obj4.alignment = True, obj4.move_limit = True user: Do not use “move_limit” and “alignmen” simultaneously for obj4. robot: obj1.alignment = True, obj4.alignment = True user: Please move all objects inside the shelf outside. robot: obj1.out = True, obj2.out = True, obj3.out = True user: I want all products of type A to be taken out from the shelf. robot: obj1.out = True user: I would like all products of type A to be taken out from the shelf without touching the products of type B and C. robot: obj1.out = True, obj2.move_limit = True, obj3.move_limit = True, obj4.move_limit = True, obj5.move_limit = True user: Bring in products from outside the shelf and arrange the same type of products. robot: obj4.alignment = True, obj5.alignment = True, obj2.alignment = True user: Arrange all products, but make sure not to take products from inside the shelf outside. robot: obj1.alignment = True, obj2.alignment = True, obj3.alignment = True, obj4.alignment = True, obj5.alignment = True, obj1.out_limit = True, obj2.out_limit = True, obj3.out_limit = True user: Retrieve all items from inside the shelf. robot:



(a) Initial state **(b) Using long / narrow hand** **(c) Using short / narrow hand** **(d) Using long / wide hand** **(e) Using short / wide hand** **(f) Goal state**

Move narrow objects to the front of the shelf. Stacked objects are carried out from the top row. Carry narrow objects off the shelf. Move wide objects to the front of the shelf. Stacked objects are carried out from the top row. Carry wide objects off the shelf.

Fig. 5. The sequence of actions obtained when given the task “Retrieve all items from inside the shelf”.

each problem, we designed three tasks and three cases of user input for each task. The results of the comparison are shown in Fig. 6. The dialogues in the figure represent the number of interactions with the user needed to achieve accurate conversion, and tokens indicate the number of tokens required to achieve accurate conversion.

From the results, it is apparent that the proposed method has achieved a success rate of approximately 90% in accurately converting tasks as intended by the user. In cases where success was achieved even without interactive communication between the user and the robot or feedback of error messages, the proposed method demonstrated success in all tasks and partially succeeded in some of the tasks that had initially failed. It was observed that in cases where the proposed method failed, the loop continued due to LLM consistently producing the same result in response to error messages.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a task planning method for robot manipulators using natural language task input. The approach involves leveraging Large Language Models (LLM) to convert natural language into symbolic representations understandable by task planners. To ensure the conversion is successful, we proposed an interactive method that allows users to confirm the conversion results and provide feedback. Additionally, to handle unexpected outputs from LLM, an error message feedback mechanism was incorporated. Also, a few-shot prompting method was used to guide the model to higher performance. Through interactive communication between users and robots and an error feedback mechanism to handle unexpected outputs from LLM, approximately 90% of 18 natural language tasks were successfully executed as intended by the users.

There are still many challenges in utilizing LLMs for task planning. The first one is the manual nature of prompt generation. The construction of prompts significantly influences the performance of LLM. Addressing this issue may require incorporating recent techniques such as Automatic Prompt Engineer (APE) [12]. The second one is the increased number of tokens LLM uses as the system becomes more complex. We believe that methods such as LLMLingua [11], which compress prompts to reduce computational costs, can

Task No.		Symbolic task	Input natural language		Non-interactive		Interactive		Interactive (err. FB)	
					Dialogues	Tokens	Dialogues	Tokens	Dialogues	Tokens
Extraction Problem	Task 1	obj1.out = True, obj2.out = True, obj3.out = True, obj4.out = True, obj5.out = True	Case 1	Retrieve all items from inside the shelf.	1	1,263	1	1,263	1	1,263
			Case 2	I want you to take out all the products from the shelf.	1	1,268	1	1,268	1	1,268
			Case 3	Robot, please remove all items from inside the shelf.	1	1,266	1	1,266	1	1,266
	Task 2	obj1.out = True, obj3.out = True, obj2.move_limit = True, obj4.move_limit = True, obj5.move_limit = True	Case 1	Please take out products of type A and type C from the shelf, but ensure that the items not being moved remain inside the shelf.	1	1,285	1	1,285	1	1,285
			Case 2	I would like products of type A and type C to be taken out from the shelf. Please do not take out any other products.	1	1,285	1	1,285	1	1,285
			Case 3	While keeping the other products inside the shelf, please move all products of type A and C from the shelf.	0	-	2	2,569	2	2,569
	Task 3	obj1.out = True, obj2.out = True, obj3.move_limit = True, obj4.move_limit = True, obj5.move_limit = True	Case 1	Please take out obj1 and obj2, and any operation on other products is prohibited.	1	1,276	1	1,276	1	1,276
			Case 2	While ensuring not to touch any other products except obj1 and obj2, please take those two items out.	1	1,280	1	1,280	1	1,280
			Case 3	I would like you to help me take obj1 through obj2 from the shelf, avoiding any contact with obj3 to obj5. Can you assist with that?	0	-	0	-	3	7,099
Arrangement Problem	Task 4	obj2.alignment = True, obj4.alignment = True, obj5.alignment = True	Case 1	I want to arrange products of the same type as those outside the shelf.	1	1,258	1	1,258	1	1,258
			Case 2	Among the three types of products, please arrange all items of the same type as those outside the shelf inside the shelf.	0	-	2	2,605	2	2,605
			Case 3	Arrange the products outside the shelf inside the shelf, and if there are products of the same type inside the shelf, please arrange those as well.	0	-	0	-	0	-
	Task 5	obj2.alignment = True, obj4.alignment = True, obj5.alignment = True, obj1.out_limit = True, obj2.out_limit = True, obj3.out_limit = True	Case 1	Please arrange all products of type B. However, make sure not to take products from inside the shelf outside.	0	-	0	-	3	5,495
			Case 2	Align the goods in B. Do not carry out obj1 to obj3 in the shelf.	1	1,283	1	1,283	1	1,283
			Case 3	As a prerequisite, we do not want all the products on the shelf to be carried off the shelf. please align all the products of type B.	0	-	0	-	0	-
	Task 6	obj1.alignment = True, obj2.alignment = True, obj3.alignment = True, obj4.alignment = True, obj1.out_limit = True, obj2.out_limit = True, obj3.out_limit = True, obj5.move_limit = True	Case 1	I want to arrange obj1 through obj4. Please ensure that products inside the shelf are not moved outside, and I would like to prohibit any operations on products that are not being arranged.	0	-	2	2,710	2	2,710
			Case 2	Arrange obj1, obj2, obj3, and obj4 without removing them from the shelf. Please refrain from manipulating other products.	0	-	0	-	5	12,263
			Case 3	It is necessary to arrange obj1 through obj4 without taking them out of the shelf. Also, please refrain from touching any other products.	0	-	3	4,150	3	4,150
-		-	-		**50%**	1,274	**72%**	1,808	**89%**	3,022

Fig. 6. Comparison of the probabilities of successfully converting natural language tasks into symbolic sequences using three methods: Non-Interactive, Interactive, and Interactive (err. FB). The green areas indicate tasks where the intended conversion was successful, while the red areas represent tasks that failed.

be employed to address this problem.

ACKNOWLEDGMENT

This research is subsidized by New Energy and Industrial Technology Development Organization (NEDO) under a project JPNP20016. This paper is one of the achievements of joint research with and is jointly owned copyrighted material of ROBOT industrial Basic Technology Collaborative Innovation Partnership.

REFERENCES

[1] M. Fox and D. Long, “PDDL2.1: An extension to PDDL for expressing temporal planning domains,” *J. Artif. Intell. Res. (JAIR)*, vol. 20, pp. 61–124, 12 2003.

[2] S. Edelkamp, “PDDL2.2: The language for the classical part of the 4th international planning competition,” 12 2004.

[3] R. E. Fikes and N. J. Nilsson, “Strips: A new approach to the application of theorem proving to problem solving,” *Artificial Intelligence*, vol. 2, pp. 189–208, 1971.

[4] T. Kawabe, T. Nishi, Z. Liu, and T. Fujiwara, “Symbolic sequence optimization approach for task and motion planning of robot manipulators,” in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pp. 1–6, 2023.

[5] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. Mccandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are Few-Shot learners,” vol. 33, pp. 1877–1901, Curran Associates, Inc., 2020.

[6] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, J. Zhou, S. Chen, T. Gui, Q. Zhang, and X. Huang, “A comprehensive capability analysis of GPT-3 and GPT-3.5 series models,” *ArXiv*, vol. abs/2303.10420, 2023.

[7] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” pp. 7871–7880, Association for Computational Linguistics, 7 2020.

[8] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. C. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. M. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, and M. Yan, “Do As I Can, Not As I Say: Grounding language in robotic affordances,” 2022.

[9] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, “Language models as Zero-Shot planners: Extracting actionable knowledge for embodied agents,” vol. 162, pp. 9118–9147, PMLR, 1 2022.

[10] S. S. Raman, V. Cohen, E. Rosen, I. Idrees, D. Paulius, and S. Tellex, “Planning with Large Language Models via corrective re-prompting,” *Foundation Models for Decision Making Workshop at NeurIPS 2022*, 1 2022.

[11] H. Jiang, Q. Wu, C.-Y. Lin, Y. Yang, and L. Qiu, “LLMLingua: Compressing prompts for accelerated inference of Large Language Models,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 10 2023.

[12] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba, “Large Language Models are human-level prompt engineers,” 11 2022.