Enhancing Task Planning Accuracy for Robots: A Fusion of Gemini AI and Large Language Models

Yeshwanth Sara
Dept. of Computer Science
University of South Florida
Tampa, Florida
yeshwanthsara28@gmail.com

***Abstract*— This paper presents an innovative approach that aims to improve the accuracy of task planning for robots. It does this by combining the capabilities of two systems - Gemini AI and large language models (LLMs). There have been concerns about the inconsistent and probabilistic nature of outputs from LLMs. To address this, our methodology uses Gemini AI to generate a single optimized task plan tree containing the most ideal sequence of steps. We then refine this initial task plan through an iterative process. This iterative refinement, along with integrating information from a vast knowledge database, allows us to enhance the accuracy and efficiency of the final task plan. When evaluated, the results show that our innovative framework outperforms previous methodologies. It demonstrates an ability to provide precise and efficient task-planning solutions tailored for robots operating in real-world environments.**

***Index Terms*-:*LLM, Gemini-AI, FOON.*

I. Introduction

Artificial Intelligence (AI) technologies have become very important for improving how robots plan their tasks and making those processes more efficient. Among these AI technologies, Large Language Models (LLMs) have shown potential, but their probabilistic (based on likelihood) nature can sometimes lead to inaccuracies in the task plans they produce. This research paper proposes an innovative new approach that uses Gemini AI, an advanced AI system, to enhance how robots plan their tasks. Gemini AI helps by refining and improving the structure of the task plans when they are in the FOON format.

Fig. 1. External View

The core of our approach is a system architecture that coordinates the interaction between the Gemini AI, the input data, and the task planning process. The process starts by taking in the input data, which typically consists of the names of dishes and lists of their ingredients. This raw data goes through some initial processing steps to clean it up and put it into a standardized format that can work well with the task planning algorithm used in the next stage. Subsequently, the refined input data is fed into Gemini AI, optimized for efficiency and effectiveness in generating task plans. Gemini AI produces a single task tree encapsulating optimized steps for preparing the dish, conforming to the FOON structure. Unlike traditional LLMs, Gemini AI ensures computational efficiency, crucial for real-time applications. After creating a task tree, we refine it by removing uncertain or expensive steps. This helps us create accurate and efficient plans. We also use a vast knowledge network to provide more contextual insights. Once we're done refining, we output the task plans in the FOON structure. This structure is easy for robots to understand and execute seamlessly. The architecture of the process ensures the components work together well, resulting in precise and efficient task plans in food format. In the following sections, we will go into deeper detail about the specifics of our methodology and explain the key components involved. We'll illustrate how all the different parts work together in a coordinated way to improve and refine the task-planning abilities for robots. Additionally, we will present the results of experiments we conducted to test our approach. The experimental data shows that our new methodology outperforms existing methods in terms of accuracy - producing more precise task plans. It also demonstrates improved efficiency in generating these accurate plans within the constraints of the FOON framework.

II. Related Work

Previous research has explored the use of object-oriented representations in robotics for various tasks. Paulius et al. [1] introduced the Functional Object-Oriented Network (FOON), a structured framework designed to enhance robotic manipulation tasks. Unlike conventional methods, FOON integrates both functional and object-oriented paradigms, providing robots with a comprehensive understanding of object interactions in complex environments. This integration enables robots to effectively manipulate objects by capturing the functional relationships between them and their corresponding actions. By emphasizing the integration of functional and object-oriented representations, FOON offers a novel approach that significantly improves robots' ability to acquire manipulation skills compared to traditional methods. This innovation represents a significant advancement in the field of manipulation learning, offering promising prospects for the development of more sophisticated and adaptable robotic systems.[2].present advancements in the Functional Object-Oriented Network (FOON) framework for robotics. This work extends the FOON model introduced in prior research, focusing on its construction and expansion capabilities. By leveraging FOON, robots can efficiently represent and reason about object interactions in complex environments, facilitating tasks such as manipulation and planning. The authors build upon their previous work and explore methods for constructing FOON structures from raw sensor data, enhancing the scalability and adaptability of the framework. This research contributes to the field of robotics by providing a comprehensive framework for representing and understanding object interactions, enabling robots to operate more effectively in diverse scenarios.[3].The method categorizes object-action relations using semantic scene graphs. Analyzing structural and semantic graph information identifies meaningful object-action pairs. This facilitates object manipulation and scene understanding tasks. The

approach enhances robotic autonomy and adaptability by leveraging object-action relationship data extracted from semantic scene graphs in complex environments. In their paper, Jain, Mosenlechner, and Beetz introduce a method to empower robot control programs with first-order probabilistic reasoning capabilities[4].This research addresses the need for robots to operate effectively in uncertain, dynamic environments. The proposed approach integrates first-order probabilistic reasoning into robotic control systems. Rather than following fixed rules, robots can reason probabilistically about potential outcomes of actions and observations. This allows them to make informed decisions by assessing likelihoods of different scenarios. Handling uncertainty through probabilistic reasoning enhances robotic adaptability and autonomy. Experiments across navigation and manipulation tasks demonstrate the utility of this method in real-world situations with inherent uncertainty. By equipping robot control with probabilistic reasoning capabilities, this work advances robotics by enabling more intelligent, adaptable systems that can reliably perform complex tasks amidst environmental unpredictability.

III. Methodology

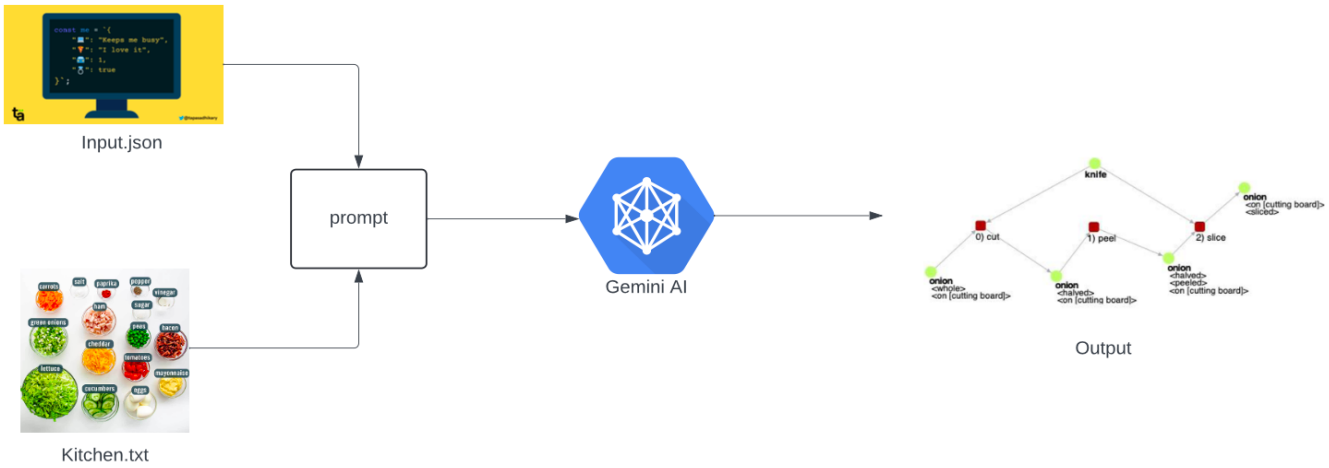

Fig. 2. Architecture

A. Data Collection and Organise the JSON Structure

Gathering the Initial Data: The first step involves collecting the raw menu data from available sources. This data includes the names of the dishes as well as the lists of ingredients for each dish. Structuring the Data in JSON Format: After gathering the raw data, it is then organized into a structured JSON (JavaScript Object Notation) format. In this format, each menu item is represented as a JSON object, which contains key-value pairs for different elements such as the dish name, its category, and the associated ingredients. For example, the provided input structure represents a menu item that falls under the "blended smoothie" category, specifically a dish called "Hash-Brown-Potato-Patty," along with the list of ingredients required to make this dish.

B. Injecting The Gemini Ai with Prompt.

After the input data is cleaned and structured into a JSON format, it is fed into Gemini AI, which is an advanced system known for efficiently generating task plans. However, Gemini AI doesn't operate on the raw input data alone. It relies on dynamic prompts that provide contextual examples of the expected JSON output for specific instructions. These dynamic prompts play a crucial role by allowing relevant information from the input JSON to be inserted into each prompt. This includes details such as dish names, required ingredients, and any available kitchen data. By incorporating this contextual information, the prompts help Gemini AI understand the specific context and requirements of the task at hand. Once Gemini AI receives the input JSON data along with these contextual prompts, it can generate task plans tailored to the given scenario. These task plans are produced in the Functional Object-Oriented Network (FOON) format, which accurately reflects the requirements and constraints outlined in the input data and contextual prompts. In essence, the dynamic prompts act as a bridge, allowing Gemini AI to understand the context of the task by including relevant details from the input data. This contextual approach enables Gemini AI to effectively process the input and generate appropriate task plans in the FOON format, tailored to the specific dish, ingredients, and kitchen environment.

C. Task Plan Refinement

After Gemini AI generates the initial task plans in the FOON format, the methodology involves a refinement stage to ensure the validity and appropriateness of the plans. This process begins by filtering the response obtained from Gemini AI. First, the system checks if the response is in the correct JSON format. If the response contains both JSON data and regular text, only the JSON portion is extracted and considered further. The system then verifies the validity of the JSON data. If the JSON is found to be invalid or incomplete, the original prompt is resubmitted to Gemini AI, and the process is repeated until a valid JSON response is obtained. This iterative approach ensures that only accurate and relevant task plans are considered for refinement. Once a valid JSON response with the task plan is obtained, the methodology proceeds to refine and improve the generated plan. The refinement process involves several steps. First, any uncertain or unnecessarily expensive steps are removed from the task plan tree to enhance its accuracy and efficiency. Next, the system leverages a vast knowledge network to provide

contextual insights and additional information relevant to the task at hand. These contextual insights are used to further enhance the quality and applicability of the task plans. By iteratively refining the task plans, validating their JSON format, and incorporating contextual knowledge, the methodology ensures that the final output consists of high-quality task plans tailored to the specific requirements of the given scenario, such as the dish to be prepared, the available ingredients, and the kitchen environment.

D. Types Of Prompts

Direct prompts are straightforward instructions or queries given to the AI system. These prompts are usually concise and direct, without much additional context or background information. Direct prompts are suitable when the task or question is simple and doesn't require a lot of contextual information.

Step-by-step instruction prompts provide a sequence of steps or instructions for the AI system to follow. These prompts are particularly useful for complex tasks that involve multiple steps or require a specific order of operations. Step-by-step prompts guide the AI through the process, breaking it down into smaller, more manageable steps.

Contextual prompts provide additional background information, context, or framing for the task or question. These prompts are often longer and more detailed, aiming to give the AI system a better understanding of the situation or problem at hand. Contextual prompts are particularly useful for tasks that require a deeper understanding of the context, such as natural language processing, question answering, or analysis of complex scenarios

E. Output

After refinement, the final task plans are outputted in the FOON (Functional Object-Oriented Network) structure. This structure represents task plans in a format easily understandable and executable by robots. It breaks down steps into actions and objects involved, capturing functional relationships between them. Using FOON allows for seamless implementation of task plans by robots in real-world scenarios, bridging the gap from high-level plans to low-level robotic actions.

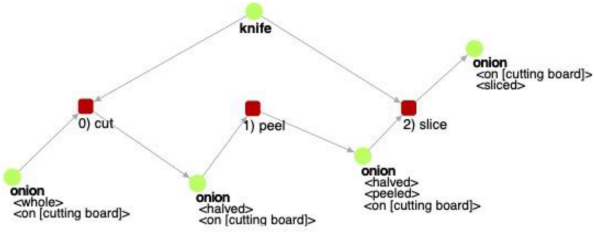

Fig. 3. Output FOON Structure

IV. EXPERIMENTS

We have conducted a series of tests using various methods of prompting, with and without the use of examples. Our findings are as follows.

TABLE I
Accuracy with different prompts

types of Prompts	Correct json files	Total json files	Accuracy
Direct prompt without example	6	33	18.18182
Direct prompt with example	16	33	48.48485
Step-by-step Instruction prompt without examples	13	33	39.39394
Step-by-step instruction prompt with examples	22	33	66.66667
Contextual prompt without examples	21	33	63.63636
Contextual prompt with examples	32	33	96.9697

Table-1 shows the accuracy achieved when using different types of prompts to generate JSON files containing task plans. The prompt types tested were:

1. Direct prompts (with and without examples)
2. Step-by-step prompts (with and without examples)
3. Contextual prompts (with and without examples)

The accuracy represents the percentage of generated JSON files that were correct out of the total JSON files produced for each prompt type. During experimentation, it was found that using direct prompts often resulted in low accuracy, with many of the JSON files containing missing or incorrect ingredients. To try to improve this, step-by-step prompts were used to provide more detailed instructions. However, challenges remained, particularly with suggesting incorrect ingredients, especially for vegetarian dish task plans.

The contextual prompts that included examples achieved the highest accuracy of 96.97%. Providing these contextual examples allowed for a better understanding of the input data, resulting in more precise task plans being generated in the JSON files.However, even with this high accuracy, there is still room for improvement when it comes to ensuring accurate ingredient suggestions, notably for vegetarian dish task plans. So, in summary, contextual prompts with examples proved most effective for accurately generating JSON task plans, but suggesting accurate vegetarian ingredients remains an area that needs further improvement.

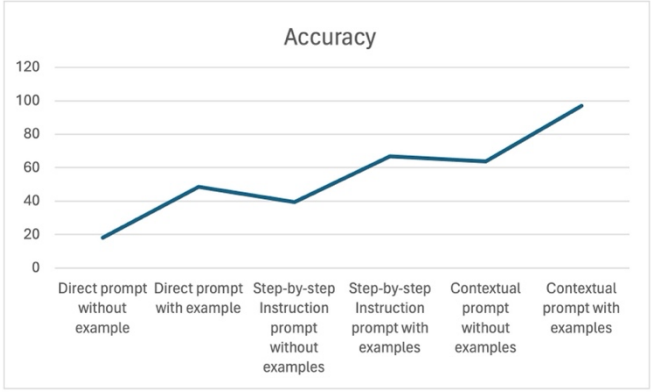

Fig. 4. Accuracy of Different Prompts

From Fig 4.1 we get to know that accuracy is more for contextual prompts with examples. It means AI is more understandable for contextual prompting.

REFERENCES

[1] Paulius, et al. "Functional Object-Oriented Network for Manipulation Learning." In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2655-2662. IEEE, 2016.

[2]Paulius, David, Ahmad B. Jelodar, and Yu Sun. "Functional object-oriented network: Construction & expansion." In 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 5935-5941. IEEE, 2018.

[3] E. Aksoy, A. Abramov, F. Worgotter, and B. Dellen. Categorizing object-action relations from semantic scene graphs. In IEEE Intl. Conference on Robotics and Automation, pages 398–405, 2010.

[4] Jain, Dominik, Lorenz Mosenlechner, and Michael Beetz. "Equipping robot control programs with first-order probabilistic reasoning capabilities." In Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, pp. 3626–3631. IEEE, 2009.