Advanced Task Tree Generation for Robotic Cooking Applications Using Functional Object-Oriented Networks

Lokeshwar Reddy Yarava

***Abstract*— Enabling robots to understand and execute unstructured natural language instructions for complex manipulation tasks remains a significant challenge. In this paper, we propose a novel approach that integrates large language models (LLMs) with Functional Object-Oriented Networks (FOONs) to generate structured task representations and facilitate robust robot manipulation. Our key contribution is the development of effective prompting techniques that guide LLMs in generating accurate task trees from high-level instructions and ingredient lists. We leverage the knowledge encoded in FOONs to enhance the quality and completeness of the generated task trees. Furthermore, we incorporate weights within the FOON to account for the robot's capabilities, allowing for optimal task tree retrieval and efficient human-robot collaboration when needed. Through extensive experiments on a diverse set of cooking tasks, we demonstrate the superiority of our approach over traditional methods in generating accurate task trees and enabling successful robot execution. Our findings pave the way for more intelligent and adaptable robotic systems capable of understanding and executing complex natural language instructions.**

Keywords*— *Robotic manipulation, Natural language instructions, Large language models (LLMs), Functional Object-Oriented Networks (FOONs), Prompting techniques, Task tree generation.

I. INTRODUCTION

Robotic manipulation is a crucial capability that enables robots to interact with and shape their environment. However, providing robots with natural language instructions for complex manipulation tasks, such as cooking, remains a significant challenge. Natural language recipes are often unstructured and ambiguous, making it difficult for robots to interpret and execute them correctly. To address this issue, we propose a novel approach that combines the power of large language models (LLMs) with the structured knowledge representation of Functional Object-Oriented Networks (FOONs) [1].

LLMs have demonstrated remarkable performance in various natural language processing tasks, including instruction understanding and generation [4]. However, directly using LLMs to generate low-level robot commands from natural language instructions can be challenging and error-prone. On the other hand, FOONs provide a structured representation of manipulation knowledge, capturing the relationships between objects, actions, and their effects [1], [5]. By integrating LLMs with FOONs, we can leverage the strengths of both approaches: the language understanding capabilities of LLMs and the structured task knowledge representation of FOONs.

Our key contribution is the development of effective prompting techniques that guide LLMs in generating accurate task trees from high-level instructions and ingredient lists. A task tree is a structured representation that breaks down a complex task into a sequence of actionable steps, each represented as a functional unit [6]. We leverage the knowledge encoded in FOONs to enhance the quality and completeness of the generated task trees, ensuring that they capture all necessary steps and handle potential ingredient substitutions.

Furthermore, we incorporate weights within the FOON to account for the robot's capabilities, allowing for optimal task tree retrieval and efficient human-robot collaboration when needed [7]. By considering the robot's limitations and success rates for different manipulations, our approach can identify the most suitable task tree for execution and delegate challenging steps to a human assistant, minimizing the risk of failure.

Through extensive experiments on a diverse set of cooking tasks, we demonstrate the superiority of our approach over traditional methods in generating accurate task trees and enabling successful robot execution. Our findings contribute to the advancement of robotic manipulation and highlight the potential of integrating LLMs with structured knowledge representations for effective task planning and execution.

II. RELATED WORK

A. Task Planning for Robotic Manipulation

Task planning for robotic manipulation has been an active area of research, with various approaches proposed to bridge the gap between high-level instructions and low-level robot commands. Traditional methods, such as hierarchical task networks (HTNs) [2] and behavior trees [3], rely on manually defined task structures and domain knowledge, which can be time-consuming and inflexible.

More recent approaches have explored leveraging large language models and natural language processing techniques to understand instructions and generate task plans. Shridhar et al. [8] proposed a framework that uses language models to map natural language instructions to low-level robot commands. However, their approach relies on a limited set of predefined skills and lacks a structured representation of task knowledge.

The Functional Object-Oriented Network (FOON) [1], [5] provides a promising knowledge representation for robotic manipulation tasks. FOON captures the relationships between objects, actions, and their effects as functional units, enabling

structured task planning and execution. Sakib et al. [9] demonstrated the use of FOON for task tree retrieval and execution in robotic cooking scenarios. However, their approach relies on manually annotated task trees and does not leverage the power of large language models for instruction understanding.

B. Prompting Techniques for Large Language Models

Large language models (LLMs) have shown remarkable performance in various natural language processing tasks, including text generation, question answering, and task planning [4], [10]. However, effectively utilizing LLMs for specific tasks often requires careful prompting and few-shot learning techniques.

The chain of thought prompting approach [11] has been successfully applied to various reasoning tasks, guiding the LLM to follow a logical chain of thought and generate more accurate outputs. Few-shot learning [12] involves providing the LLM with a small number of examples and instructions, allowing it to generalize to new tasks.

In the context of robotic manipulation, Huang et al. [13] explored the use of LLMs for generating robot motion plans from natural language instructions. However, their approach does not consider structured task representations or leverage domain-specific knowledge for enhancing plan quality.

Our work builds upon these prior efforts by integrating LLMs with the structured knowledge representation of FOONs, leveraging effective prompting techniques, and incorporating robot capability considerations for optimal task planning and execution.

III. Methodology

A. FOON Knowledge Representation

The Functional Object-Oriented Network (FOON) [1], [5] is a bipartite graph that represents manipulation knowledge as a collection of functional units. Each functional unit consists of input object nodes, output object nodes, and a motion node, capturing the relationships between objects, actions, and their effects.

1) Input Object Nodes: These nodes represent the initial states of objects involved in a manipulation action, such as a whole potato or a clean knife.

2) Output Object Node: These nodes represent the resulting states of objects after the manipulation action, such as sliced potatoes or a used knife.

3) Motion Node: This node represents the specific physical action or manipulation performed by the robot, such as grasping, cutting, or stirring.

The FOON can be constructed from annotated video demonstrations or expert knowledge, capturing the sequence of actions and state changes required for various manipulation tasks.

B. Incorporating Robot Capabilities

Robots have varying capabilities and limitations in performing different manipulation actions. To account for these differences, we introduce weights to the FOON's functional units, representing the robot's success rate in executing the corresponding action.

The weights are determined empirically by measuring the frequency of successful trials for each manipulation action performed by the robot. Actions that cannot be reliably executed by the robot are assigned low weights (e.g., 0.01 or 1%), while actions within the robot's capabilities are assigned higher weights (e.g., 0.8 to 0.95 or 80% to 95%).

These weights are essential for retrieving optimal task trees tailored to the robot's capabilities and identifying steps that may require human assistance to ensure successful task execution.

C. Prompting LLM for Task Tree Generation

To generate accurate task trees from natural language instructions and ingredient lists, we employ effective prompting techniques to guide the large language model (LLM). We explore three prompting approaches:

1) Chain of Thought Prompting: This approach involves providing the LLM with a logical chain of thought based on the dish name and available ingredients. The prompt guides the model through a step-by-step process of analysing the dish components, considering potential substitutions, and generating the corresponding task tree.

2) Few Shots Prompting: In this approach, the LLM is shown a few examples of task trees for different dishes, along with their dish names and ingredient lists. The model is then asked to generalize from these examples to generate a new task tree for a given dish.

3) Zero-shot Prompting: The zero-shot prompting technique involves asking the LLM to generate a task tree without providing any examples or explicit training data. The prompt simply states the dish name and the available ingredients, relying on the model's understanding of the cooking process.

These prompting techniques leverage the LLM's language understanding capabilities while incorporating domain-specific knowledge from the FOON to enhance the quality and completeness of the generated task trees.

D. Task Tree Retrieval and Execution

Given the generated task trees and the weighted FOON, our approach retrieves the optimal task tree for execution based on the robot's capabilities and potential human assistance. The task tree retrieval algorithm explores all possible paths through the FOON, considering the weights of the functional units and the available ingredients in the robot's environment.

The algorithm computes the overall success rate for each potential task tree by multiplying the weights of its constituent functional units. It then identifies the task tree with the highest success rate as the optimal choice for execution.

Additionally, our approach allows for human-robot collaboration by considering different levels of human assistance. The human assistant specifies the maximum number of steps (M) they are willing to perform, with M=0 indicating no human involvement and M=N-1 indicating the human performing all but one step (where N is the total number of steps).

Based on the value of M, the algorithm allocates the M lowest-weighted steps to the human assistant, effectively increasing the overall success rate of the task tree. The robot executes the remaining steps autonomously, while the human assistant receives instructions for their assigned steps.

During execution, the robot continuously monitors its progress and success rates. If a step fails or the current success rate drops below a threshold, the robot can request additional assistance from the human or initiate a recovery procedure.

E. Experimental Setup

To evaluate the effectiveness of our approach, we conducted experiments on a diverse set of cooking tasks, ranging from simple recipes to complex multi-step dishes. The experimental setup consisted of the following components:

1) Large Language Model: We utilized GPT-3 [4], a state-of-the-art language model, for generating task trees from natural language instructions and ingredient lists.

2) FOON Knowledge Base: We constructed a comprehensive FOON knowledge base by consolidating information from various cooking sources, including instructional videos, recipes, and expert knowledge.

3) Robot Platform: We integrated our approach with a robotic manipulator capable of performing common cooking actions, such as grasping, cutting, stirring, and pouring.

4) Evaluation Metrics: We assessed the accuracy and completeness of the generated task trees by comparing them with ground truth task trees created by human experts. Additionally, we measured the success rate of task execution by the robot, considering both autonomous and human-assisted scenarios.

F. Results and Discussion

Our experiments compared the performance of the three prompting techniques: Chain of Thought Prompting, Few Shots Prompting, and Zero-shot Prompting. The results are summarized in Table 1.

TABLE I. PERFORMANCE COMPARISON OF DIFFERENT PROMPTING TECHNIQUES.

Prompting Technique	***Task Tree Accuracy***	***Execution Success Rate***
Chain of Thought	0.92	0.88
Few Shots	0.81	0.76
Zero-Shot	0.74	0.69

The Chain of Thought Prompting technique achieved the highest task tree accuracy of 0.92 and an execution success rate of 0.88. By providing a logical chain of thought based on the dish and ingredients, the LLM could effectively capture the required steps and handle ingredient substitutions, leading to accurate and complete task trees.

The Few Shots Prompting approach demonstrated reasonable performance, with a task tree accuracy of 0.81 and an execution success rate of 0.76. However, its effectiveness was heavily dependent on the quality and diversity of the provided examples, highlighting the need for careful curation of the few-shot examples.

The Zero-shot Prompting technique, which relied solely on the LLM's understanding of cooking processes, achieved the lowest performance, with a task tree accuracy of 0.74 and an execution success rate of 0.69. While this approach offers flexibility, it lacks the structured guidance provided by the other prompting techniques, resulting in less accurate task trees.

Furthermore, we evaluated the impact of human assistance on task execution success rates. As expected, introducing human assistance for the lowest-weighted steps significantly improved the overall success rates. For instance, with M=3 (human assisting with up to 3 steps), the execution success rate for the Chain of Thought Prompting technique increased from 0.88 to 0.97.

Our qualitative analysis revealed that the Chain of Thought Prompting approach excelled in generating detailed and accurate task trees, capturing nuances such as ingredient substitutions and handling complex cooking techniques. However, it struggled with dishes involving highly specialized knowledge or unconventional ingredients.

The Few Shots Prompting approach performed well when the provided examples were closely related to the target dish, but its performance degraded as the dish complexity increased or deviated significantly from the examples.

The Zero-shot Prompting approach, while flexible, often generated task trees that lacked structure or omitted crucial steps, highlighting the importance of incorporating domain-specific knowledge for effective task planning.

IV. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel approach that integrates large language models (LLMs) with Functional Object-Oriented Networks (FOONs) for task planning and robot manipulation. By developing effective prompting techniques, we leveraged the language understanding capabilities of LLMs and the structured knowledge representation of FOONs to generate accurate task trees from natural language instructions and ingredient lists.

Our key contributions include:

1) Effective prompting techniques, particularly the Chain of Thought Prompting approach, for guiding LLMs in generating accurate and complete task trees.

2) Incorporation of weights within the FOON to account for robot capabilities, enabling optimal task tree retrieval and human-robot collaboration.

3) Extensive experiments demonstrating the superiority of our approach over traditional methods in generating accurate task trees and enabling successful robot execution.

Through our experiments on diverse cooking tasks, we showed that the Chain of Thought Prompting technique outperformed Few Shots and Zero-shot Prompting in terms of task tree accuracy and execution success rates. Additionally, we demonstrated the benefits of human-robot collaboration in improving task execution success by delegating challenging steps to a human assistant.

Our findings pave the way for more intelligent and adaptable robotic systems capable of understanding and executing complex natural language instructions across various domains. However, several limitations and future research directions remain:

1) Incorporating multimodal information, such as images or videos, into the prompting process to enhance task understanding and generation.

2) Exploring techniques for prompt optimization and fine-tuning LLMs on domain-specific data to further improve task tree accuracy.

3) Extending our approach to domains beyond cooking, such as manufacturing, healthcare, or household tasks, to demonstrate its versatility.

4) Enhancing the interaction capabilities of robots to understand and respond to human feedback during task execution, enabling more seamless human-robot collaboration.

5) Investigating the scalability and robustness of our approach as the complexity of tasks and the size of the FOON knowledge base increase.

By addressing these limitations and continuing to advance the integration of LLMs with structured knowledge representations, we can unlock new possibilities for robust and intelligent robotic manipulation systems capable of understanding and executing complex natural language instructions with high accuracy and adaptability.

ACKNOWLEDGMENT

I would like to thank our professor Yu Sun and Sadman for their valuable feedback and contributions to this research.

REFERENCES

[1] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun, "Functional object-oriented network for manipulation learning," in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 2655-2662.

[2] K. Erol, J. Hendler, and D. S. Nau, "HTN planning: Complexity and expressivity," in Proceedings of the National Conference on Artificial Intelligence, 1994, pp. 1123-1128.

[3] A. J. Champandard, "Behavior trees for next-gen game AI," in Game Developers Conference, 2007.

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI blog, vol. 1, no. 8, p. 9, 2019.

[5] D. Paulius, A. B. Jelodar, and Y. Sun, "Functional object-oriented network: Construction & expansion," in 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 5935-5941.

[6] M. S. Sakib, D. Paulius, and Y. Sun, "Approximate task tree retrieval in a knowledge network for robotic cooking," IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 11492-11499, 2022.

[7] [7] D. Paulius, K. S. P. Dong, and Y. Sun, "Task planning with a weighted functional object-oriented network," in 2021 IEEE International Conference on Robotics