Enhancing Robotic Task Execution in Dynamic Environments: The Development and Evaluation of Task Trees for Improved Kitchen Automation

line 1: 1st Given Name Surname   
line 2: *dept. name of organization (of Affiliation)*  
line 3: *name of organization (of Affiliation)*  
line 4: City, Country  
line 5: email address or ORCID

*Abstract*— This paper tackles the problem of efficiently teaching robots to carry out activities in the kitchen, which is a setting prone to sudden changes and potential mistakes. I present the idea of a "task tree," which is basically a structured manual that guides robots through the stages that must be followed in order to complete a given task. I investigate ways to produce these task trees from natural language instructions using Google's AI tool, Gemini. This study assesses the effectiveness of three different methods for producing task trees that are understandable and useful. The most organized and thorough method considerably improves the robots' comprehension and performance of cooking responsibilities, according to the results. Through this research, robots in dynamic contexts like kitchens will be safer and more competent partners.

Keywords— Robotic Task Execution, Natural Language Processing, Task Trees, Kitchen Automation, **Automated Task Guidance**

# Introduction

It can be challenging to teach robots how to execute assignments, particularly in dynamic areas like kitchens where everything can change rapidly, and mistakes might have serious consequences. Since human settings are unpredictable, traditional methods of programming robots—which frequently depend on strict, planned sequences of actions—don't function well in these kinds of environments. Because they are designed to do exact and routine tasks, robots frequently misinterpret the more sophisticated and frequently imprecise instructions that humans give. This discrepancy may result in ineffectiveness or, worse, mistakes that interfere with the work at hand and potentially cause accidents.

It is still very difficult to comprehend and understand human instructions in a way that is compatible with robotic capabilities. Robotic systems require instructions that are more structured and obvious than those written by humans.

For instance, a basic task like "prepare a salad" includes a lot of underlying choices and subtasks, such as choosing fresh ingredients, cleaning and chopping vegetables, and mixing them in the right amounts. These are all tasks that can differ significantly from one execution to the subsequent execution. To address these challenges, my research presents an innovative approach: the "task tree." A task tree is like a map guiding a journey through unfamiliar regions: it's basically an organized handbook for robots. It decomposes a task into discrete, sequential steps that define each action the robot must perform and the setting in which it must perform them. This methodical technique improves the robot's capacity to both precisely follow instructions and dynamically adjust to changes in the task variables or environment.

This work builds these task trees from natural language instructions using Gemini, an advanced AI tool developed by Google. Gemini analyzes the complexity and context of human language by utilizing cutting-edge machine learning methods, particularly in the field of natural language processing (NLP). I enable Gemini to generate precise, detailed task trees which guide robotic actions in a kitchen environment by providing it with a wide range of datasets that cover a range of cooking settings and instructions.

More specifically, my work investigates how Gemini can convert ambiguous and occasionally inadequate human instructions into precise and workable plans that robots can adhere to. I examine the performance of three different task tree generation techniques, contrasting their capacity to yield comprehensible, practical instructions for robotic action.

The most comprehensive and detailed of these techniques has demonstrated promise to greatly enhance the robots' comprehension and execution of cooking tasks. These task trees not only assist in the completion of the current work, but they further our knowledge of human-robot interaction and open the door to the development of more naturalistic and intuitive techniques for robots to interact with humans.

Our ultimate objective is to assist robots in learning and executing tasks more effectively, making them safer and more helpful partners in our daily lives. Task trees can increase the reliability and efficacy of robots as partners in not just kitchens but potentially in any dynamic environment by increasing task execution precision and reducing the likelihood of errors.

This research opens up new possibilities for the future of automation in daily life, where robots and humans coexist peacefully and safely, in addition to contributing to advances in the field of robotic task performance.

# Related Work

Robotic system integration in dynamic environments, such as kitchens, requires efficient job planning and execution methodologies that can adjust to sudden changes and chaotic circumstances. This section highlights key research that deepens our knowledge of robotic task planning utilizing intelligent systems and structured knowledge representations.

## Test Planning and Knowledge networks

Utilizing Large Language Models (LLMs) coupled with knowledge networks has been one of the most significant developments in robotic task planning for kitchens. In a recent paper, Sakib and Sun provide a novel method for improving planning accuracy and job efficiency for robotic cooking using a task tree generating pipeline. Using LLMs, their approach retrieves recipe instructions and transforms them into structured task trees. A retrieval procedure then refines the results by eliminating unreliable nodes and choosing paths that are most efficient [1]. A strong foundation for managing the many and intricate requirements of cooking jobs is provided by this methodology, which considerably advances task planning accuracy and efficiency.

## Functional Object-Oriented Networks (FOON)

Functional Object-Oriented Networks (FOON) offer a significant improvement in robotic manipulation learning. FOON is a graphical model that was first presented by Paulius et al. [2] that illustrates the relationship between motions of functionally linked items and their connectedness. Robots are able to produce precise manipulation motion sequences and understand task objectives with the help of this organized knowledge representation. It has proven especially helpful in situations requiring accurate and adaptive task performance, showcasing the adaptability and practicality of knowledge-based robotics systems. Developing and Expanding FOONs: Paulius, Jelodar, and Sun investigated strategies for developing and enlarging FOONs by incorporating information from diverse instructional videos and building on the fundamental idea of FOON [3]. This growth makes it possible to build a more resilient and extensive network that, by utilizing object similarities and widening object categories, can adjust to new scenarios. These developments improve the adaptability and versatility of the network, allowing it to handle new activities and surroundings that weren't previously encoded in the system.

## Comparative Analysis of task planning methods

The combination of LLMs with FOON signifies a substantial advancement in the creation of more precise and adaptive robotic systems for kitchen tasks. These techniques not only make it easier to do tasks precisely, but they also allow the cooking environment to be modified as necessary. Research has demonstrated that the generative powers of LLMs in conjunction with structured knowledge representations like FOON significantly improve robotic task planning's accuracy and efficiency [1], [3].

**Challenges and the future:** Despite these developments, there are still a lot of issues to be resolved, especially with regard to system dependability and the capacity to adapt to different types of kitchen configurations. Future studies should concentrate on strengthening these systems' resilience to make sure they can withstand the unpredictable nature of real-world kitchen settings. Furthermore, broadening the knowledge pool to include a more diverse range of cooking techniques and components would augment the system's suitability and efficiency.

**In summary**, an important step in the field of robotic cooking has been made with the creation of intelligent task planning systems that make use of FOON and LLMs. These solutions guarantee that robots can carry out cooking jobs with increased accuracy and flexibility in addition to streamlining the task tree generation process. This corpus of work establishes a solid basis for continued study into the integration of more complex cognitive models into robotic systems, with the goal of improving the systems' interaction and functionality in human-centered environments.

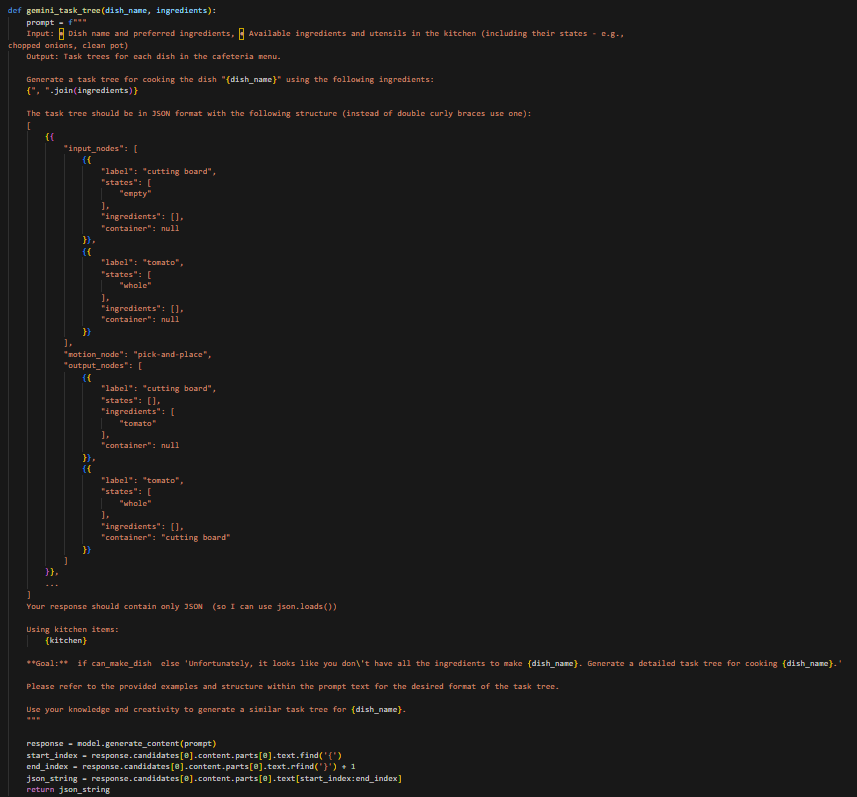
# Methodology

The task tree creation process structures complex tasks into sequential, manageable steps for robotic execution. It starts by breaking down a main task into subtasks, each defined by specific actions and outcomes, using natural language processing tools like Google's AI tool Gemini to translate instructions into structured steps. Each element of the tree is designed to minimize ambiguity and enhance clarity, crucial for accurate robot performance in dynamic settings like kitchens. A Machine learning based approach enables the task tree to learn from past executions, improving accuracy and adapting to changes in the environment through real-time data, thus eventually refining the robot's actions dynamically.

In this work, the effectiveness of three different approaches to task tree generation for robotic kitchen tasks is examined. These techniques were created to decipher and organize natural language instructions into a format that kitchen robots might use. Every method seeks to translate these commands into comprehensive task trees that lead the robot through the actions required to finish cooking assignments, the following are the approaches used in this research.

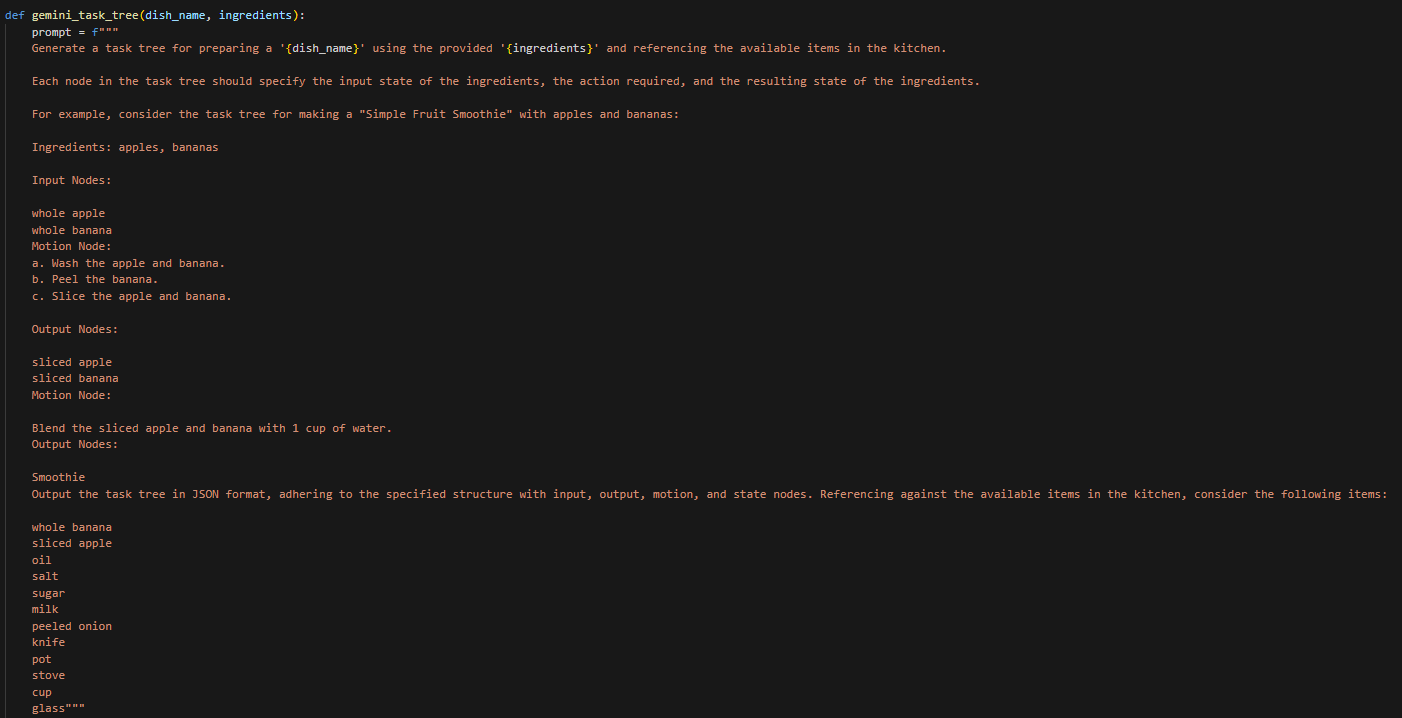
## A comprensive cooking task tree

This methodology guides the task tree development process with well-crafted and organized suggestions. The prompts list the dish's name, recommended ingredients, and the states of the various kitchenware that are available. With detailed descriptions of input nodes (beginning conditions), motion nodes (necessary actions), and output nodes (anticipated outcomes after the action), the method seeks to produce a task tree in a JSON format. Clarity and detail are prioritized in this method to reduce uncertainty and guarantee precise robot execution.

Fig. 1. Approach A prompt

## Ingredient-based Task Tree Generation

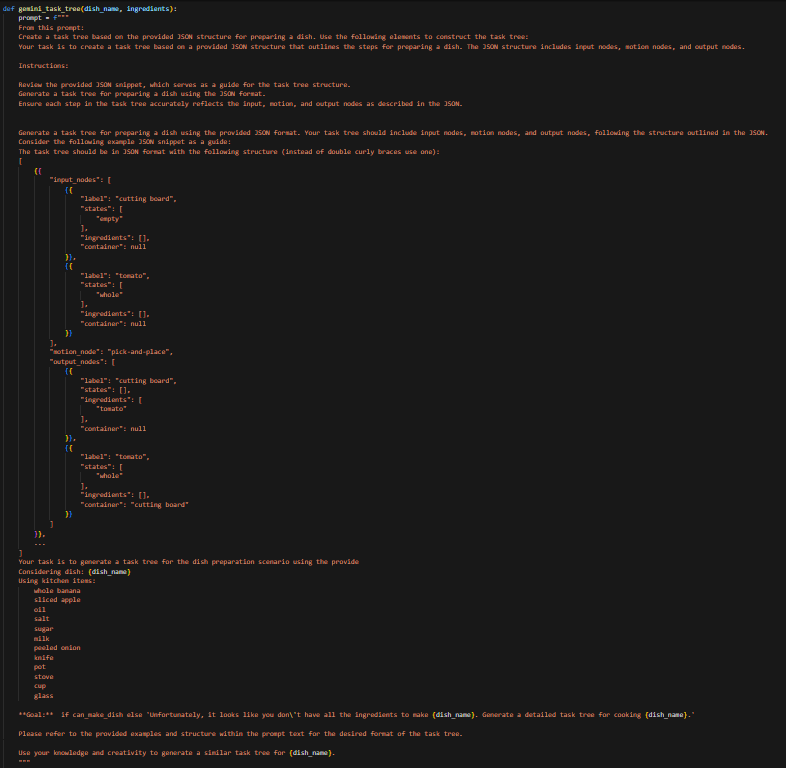
The second method is to create task trees using the mentioned kitchen objects and the ingredients that are on hand. In contrast to Approach A, this methodology employs structured prompts that provide an overview of the task tree. The task sequence is primarily determined by the ingredients that are now accessible, necessitating the robot to modify its activities in accordance with available resources.

Fig. 2. Approach B Prompt

## Vague JSON-based Task Tree Creation

The least structured approach is this one, in which task trees are created using a given, but ambiguous, JSON structure. There is less advice provided by this method when building the task tree, which increases output variability. The JSON structure outlines the elements of the task tree that are required, but the execution phase is mostly responsible for handling the details and contextual adaptation.

These approaches are theorized to offer varying levels of effectiveness, with the hypothesis that greater specificity in task prompts will result in more accurate and efficient robotic task execution.

Fig. 3. Approach C Prompt

# Expiermentes/Discussion

**Experimental Setup:** The studies were carried out in a simulated kitchen setting with a robotic system that was trained to carry out cooking chores using task trees produced by three different approaches. The robot's tasks included using a variety of culinary utensils and ingredients to cook meals in accordance with the cafeteria menu. The correctness of the task execution and the robot's operational efficiency were the evaluation metrics for each technique, with a focus on speed and adherence to the given task instructions.

**Validation Process:** Every set of task trees produced by the approaches underwent a rigorous set of tests. These assessments assessed how well the robot performed in carrying out the tasks with accuracy and efficiency. Anomalies in performance were observed, and methodical modifications were implemented to enhance every strategy. The goal of this iterative approach was to decrease errors and speed up task completion, enabling the robot to perform tasks of diverse complexity and unpredictable nature.

## *Detailed Results for the Approaches*

**A. Comprehensive Cooking Task Tree**

The highest levels of operating efficiency and precision were obtained using this strategy. The task trees that were constructed were specific and had well-defined phases, which made it possible for the robot to complete jobs with little uncertainty. The approach's efficacy in high-precision contexts was demonstrated by the high success rate in performing complex dishes with the use of structured and precise suggestions.

**Example Scenario**: The robot successfully coordinated the chopping, mixing, and cooking steps of a multi-ingredient dish without the need for human assistance.

**B. Ingredient-based Task Tree Generation**

This method worked rather well for less complicated recipes, but it had trouble with more complicated ones. Sometimes inefficient activities were taken in the absence of clear directions, such as handling substances unnecessarily. **Example Scenario:** The robot fared well when preparing a simple salad, but the task trees did not offer enough detail to ensure optimal performance when preparing a multilayer cake, which required precision timing and operations.

**C. Vague JSON-based Task Tree Creation**

The robot's capacity to comprehend and complete the tasks varied significantly under this technique, resulting in the most variable performance. The ambiguous cues frequently resulted in inefficiencies and task execution mistakes that needed manual fixes.

**Example Scenario:** Simple activities, like boiling pasta, were done well enough, but more difficult jobs, such making a stew that needed to cook over several hours, were done badly.

## *Comparative Analysis and Discussion*

The task tree generation process benefits greatly from prompt specificity and structural clarity, as demonstrated by the experiments. **Approach A** performed better than the others, particularly in complicated cooking tasks, confirming the theory that precise and organized cues greatly improve robotic performance. On the other hand, Approaches **B** and **C** showed how a decline in prompt specificity could result in a drop in productivity and a rise in operational errors, which would affect the overall usefulness of robotic helpers in a kitchen environment.

## *Implications for Robotic Task Planning*

These results have broad implications for automated system design in dynamic contexts requiring high degrees of adaptability. The effectiveness of Approach A indicates that operational difficulties in automated systems can be significantly reduced by careful programming and thorough task design preparation. This knowledge is essential for creating autonomous robotic systems in challenging environments in the future.

## *Future Directions*

Additional investigation should concentrate on incorporating machine learning methodologies to augment the robot's capacity to incorporate knowledge from prior assignments and elevate its operational efficiency in real-time. Furthermore, adding sensory feedback and increasing the task tree complexity can enable robots to adapt their behavior in real time in response to changes in their surroundings, creating robotic systems that are more intelligent and responsive.

##### References

1. M. S. Sakib and Y. Sun, "From Cooking Recipes to Robot Task Trees--Improving Planning Correctness and Task Efficiency by Leveraging LLMs with a Knowledge Network," arXiv preprint arXiv:2309.09181, 2023.
2. D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun, "Functional object-oriented network for manipulation learning," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2655-2662, IEEE, 2016.
3. D. Paulius, A. B. Jelodar, and Y. Sun, "Functional object-oriented network: Construction & expansion," in IEEE International Conference on Robotics and Automation (ICRA), pp. 5935-5941, IEEE, 2018.