A Comparitive Analysis of Prompting Techniques For Task Tree Generation using LLM

Gayathri Priyanka Namburu
Department of Computer Science and Engineering
University of South Florida
Tampa, USA
gayathripriyanka@usf.edu

***Abstract*—Robots find it difficult to follow natural language instructions since they are unstructured. This paper investigates the effectiveness of algorithm for task tree construction based on the data provided by the large language model in the cooking domain, which represent the sequential actions needed to cook specific dishes. Task trees provide a structured representation of the actions necessary to realize an objective, which makes them the basics for both robotic task planning and execution. Nevertheless, the translation of natural language commands into correct task trees is still a major obstacle. We explore and evaluate three different prompting approaches and evaluate their performance on a menu, analyzing the accuracy and limitations of each approach. Through experiments and reviews from peers, I showed the superiority of a prompting approach in the production of correct and consistent task trees. This project brings a novel idea of human-robot integration by connecting the traditional human instruction method with robot understanding. The results of this study may, in turn, have the capacity to influence not only cooking but many other fields as well, paving the way for the next generation of intelligent robotic assistants who may be able to comprehend and carry out complex tasks based on human instructions.**

Index Terms - Task tree generation, large language models, prompt engineering, robotic cooking, , functional object-oriented network (FOON).

I. Introduction

In the field of robotics, natural language commands have been a major challenge for robots because of their informal nature. This problem can be overcome by designing a reliable knowledge representation that will put robots in a position to be able to understand and properly implement the instructions. An example of such representation is a task tree, which is used to define the steps that need to be taken towards reaching the goal. A task tree is an enumerated flow of functions that a robot is expected to perform; a functional unit represents each function. These action units contain the input nodes, which will describe the initial state of the objects; the output nodes, which will show the expected state after a movement has been performed; and the motion node, which will specify the physical movement the robot needs [1][6]. The primary role of task trees is in their function as the bridge between the human natural language and the robot's understanding, which in turn facilitates the more effective execution of tasks and reduces the room for errors. Task tree generation using large language models is the primary focus of this project, showing its application to the cookery domain. The objective here is to impact the power of language models and prompt engineering in the design of effective knowledge representation approaches for the process of generating exact task trees from natural language instructions, which later aids in the development of a reliable knowledge representation for robots in the culinary context [5][6]. However, in dynamic real-world environments, encountering entirely new tasks or variations on existing ones is inevitable [2].

II. Related Work

A. LLM with FOON

In the past few years, it has been shown that integrating LLMs within the framework of functional object-oriented networks (FOON) is a promising approach to robotic task planning. Sakib and Sun [2] have introduced an approach that enables crisp plan tree optimization for robots constructed utilizing the LLMs. They are trying to stabilize the plans provided by the output of the LLM by taking the main points and objects from the output and merging them into a single task tree. Such a consolidation mechanism is an effective tool to limit the emergence of differences and erred outputs due to the inherent variability of LLM outputs. This, in turn, enhances the overall robustness and applicability of the generated plans in complex robotics scenarios. The presented approach demonstrates the possibility of providing initial virtual robotic task plans based on LLMs, and the consolidation process further brings them to the reality of actual robotic plans.

A separate field of research investigates into task tree generation utilizing large language models. Apart from this, notable research by Sakib and Sun [2] that explains the use of LLMs with knowledge networks for the purpose of changing cooking recipes into robot task trees has proved to be effective. LLMs' linguistic understanding skills embedded along with a tailored FOON representation was the tool they used to build a system that translates natural language recipes into task trees. This approach will illustrate that powerful tools are created when LLMs come together with domain-specific knowledge, increasing the speed and accuracy of robotic task planning, even in the culinary domain. The strength of this system to convert human-readable recipes to a robot task tree opens a new gateway to kitchen robot automation and supporting kitchen tasks. In an (LLMs + FOON) context, the successful integration of NLP with structured knowledge representation schemas illuminates the synergistic benefits of this combination for robotic operations.

B. Enhancing Robotic cooking through LLM and FOON

The application of language models onto FOON has resulted in some positive outcomes, which let robots learn and understand advanced tasks in cooking. Sakib et al. [3] proposed an innovative approach of approximate task tree extraction in a knowledge network constructed for the automation of cooking jobs. Their approach involves the use of a language model that encodes both the query and the task trees into a semantic space, which helps for efficient and accurate task tree retrieval at the given query. By comparing the embeddings, the system can determine the task trees that lay close to the query embedding so that the job is automatically well-matched to the specific cooking task in hand.

In a way, this approach represents a huge revolution in research. It allows robots to quickly and accurately get what is needed from a big knowledge base; hence, the more suitable robots are for working in many cooking situations. The language model's capacity to learn the semantic relationship between query and task tree is the most important, though the system, based on its semantic similarity, can best identify the most relevant task trees. Addressing the use case of FOON here, the integration of language models with semantic understanding capabilities is a good example of how natural language becomes an indispensable part of robotics to deliver the required performance and flexibility in complex areas like cooking. In essence, this renders the processing of such instructions seamless, ensuring collaboration between robots and humans that can carry out multiple cooking tasks by natural language.

C. Background of FOON

The FOON, a Functional Object-Oriented Network framework, recently emerged as a robust knowledge representation model that mirrors the complex associations of objects, operations, and effects within an action in a task-based manner. The foundation work by Paulius and others [4] presented FOON as the technique for encoding manipulative knowledge for robots during learning. The main components of FOON are the functional units, which express input and output values for objects and their actions in the current state. This analysis serves the purpose of breaking down complex problems into a series of functional units that make it possible for robots to reason out the results of their actions and make relevant decisions later. FOON provides the ordered environment from which robots can learn tasks easily by representing dependencies and interactions among objects and actions, which can be essential to completing certain tasks.

Paulius and his co-workers [5] have expanded the FOON concept even further by not only outlining but also proposing methods for its construction and scalability. Thus, they created the learning system, which was able to construct FOON from demonstrations and expand the FOON network by acquiring new knowledge. Furthermore, this revolutionary work built the base for the scalability and adaptability of FOON, letting it expand and get used to the emergence of new tasks or entities. One key feature of developing intelligent and adaptive robotic systems that can undertake multiple tasks in variable environments is the capability of FOON to get constantly updated and gradually improved based on new experiences and observations. The pragmatic and tangible approach of Paulius et al. [5] contributes to the usefulness of grid representation FOON for application in robotic manipulation scenes, turning it into an effective tool for manipulation knowledge organization.

In the subsequent study, Paulius et al. [6] investigated the usage of weighted FOON for implementing task planning. They came up with an original weighting ratio that would allow some key input units to dominate the calculations depending on the task at hand. Such an approach provides extra precision to the robot so it can prioritize and follow a more directed plan towards a final goal. Consequently, the process of planning emphasizes the unit functions that are considered critically important for accomplishing the task victoriously. The FOON approach, which takes the weight on itself, marks the next step in robotic task planning. It enables robots to make better decisions and distribute resources more efficiently, ultimately accomplishing better results and improving task performance measures.

The Sakib et al. [6] study was a deep evaluation of the menu created by FOON. They appraised the quality and implementation of the obtained recipes by comparing them with recourses written by people. They confirmed FOON's ability to capture the key actions and ingredients needed for recipe performance, giving evidence of its appropriateness for creating recipes of the appropriate type and quality. The capability to generate assembling instructions that are almost like those created by humans is proof of the power and expressiveness of FOON encoding. This work does not only have a symbolic meaning in the academic setting but also outlines the applicability of FOON in real life, especially within the scope of robot cooking, where the production of secure and usable recipes is a crucial issue. The Sakib et al. [6] study, which aimed at evaluating the FOON-based recipe creation, has offered invaluable lessons about its strengths and weaknesses, thus paving the way for more improvements and refinements in this area.

Language Models have opened new avenues for automatic task tree generation. Research by [3] explores the application of fine-tuned models to translate recipes directly into task tree structures. This approach leverages the inherent ability of LLMs to understand and process sequential instructions. However, the inherent limitations of LLMs, such as factual errors or nonsensical outputs, can lead to inaccuracies in the generated task trees. Our work addresses these limitations by proposing a multi-step pipeline that utilizes a fine-tuned LLM for initial recipe generation, followed by a dedicated fine-tuned model to convert the instructions into a structured task tree format. This two-stage approach aims to harness the strengths of LLMs for understanding natural language instructions while mitigating the risk of errors through dedicated model training on a recipe-task tree dataset.

III. METHODOLOGY

A. Prompt Engineering with Gemini

The technique utilized for this study involved a large language model, Gemini, and prompt engineering to come up with task trees for several dishes. This roadmap was created by carefully designing prompts that could have accomplished the purpose of instructing Gemini in interpreting and engineering accurate task trees. Examples of task tree generation were via prompts that Gemini could learn and understand. The task finally becomes much easier over time.

We investigated three main prompting approaches to explore their effectiveness in guiding Gemini. These processes are targeted to provide Gemini with the comprehension to generate precise task trees for any dish as the occasion may be. Each tree is generated separately, and it is written in JSON files. Prompt engineering can be represented in the form of Structured prompting, open-ended prompting and zero-shot prompting. Structured prompting implies creating a plan with certain rules and outlines of a tree for task execution.

B. Structured Prompting

Structured prompting facilitates stepwise navigation and provides the language model with given examples, resulting in the model's ability to follow a systematic format. It states the exact process of the task tree development, ingredients, actions, and the desired structure. This approach is well-suited and the best approach to make task tree generation easier as it can offer directions in a strict sense of order to the language model. Such an approach involved the application of master prompts that served as a bridge between natural linguistic descriptions of tasks and the robotic form of understanding.

The first approach, structured prompting, provided Gemini with a pre-defined template for the task tree structure. This template included the instruction for the model along with an example of the output structure which describes the placeholders for different elements like the initial state of ingredients mentioned as input nodes, desired outcomes mentioned as output nodes, and the specific actions required for each step of cooking process mentioned as motion nodes. We populated these placeholders with examples from the cafeteria menu assigned, offering context, and illustrating the desired format for the task tree.

C. Open-Ended Prompting

Unlike a regular prompt, which is characterized by a clear structure, open-ended prompting allowed room for interpreting and writing the cooking experience according to individual judgment and abilities. For this method, the model set approaches an analysis of the recipe and can make multiple versions and creative interpretations [9]. Thanks to that, it emphasized how the cooking process should reflect the presentation. It included little personal touches and alterations that may differ from the "common" or "correct" techniques that could be applied during cooking.

The language model was applying inventiveness, and the emotions were being sharply conveyed despite the givenness of the unstructured text. Indeed, this individualism of cooking emphasizes the art of cooking, where people use their preferences and techniques to make different dishes. As the language model can comprehend and describe the cooking process from free-form expressions, the methodology was developed to use the model's ability to understand the process and represent it flexibly and adaptable while still capturing the essential underlying relationships and operations using FOON.

D. Zero-Shot Prompting

The zero-shot prompting strategy was a different type of challenge for the language model since it needed the model to construct the task tree without specific sampling task data. In the present case, the prompts did not have specially stated instructions but were dependent on the language model's general capabilities and knowledge. Despite the apparent cons of this approach, it worked as a starting point for distributing knowledge to general tasks compared to the more elaborate and specified training.

The methodology used here is a zero-shot approach to assess the zooming capability and its limitation on generalizing the language model for tree generation classes between cooking tasks [6]. This approach argued that large language models would, in all probability, employ their general comprehension and knowledge as a resource for handling unspecified situations even without acquiring specifically targeted training data.

Some of the additional prompting techniques implemented by my peers are One-shot prompting, in which a task is performed using a model with only single example or prompt. Few- shot prompting, where a model is trained with small number of examples for performing a particular task.

By comparing these contrasting approaches, we aimed to evaluate the impact of varying levels of guidance on the quality of the generated task trees. We analyzed factors such as the clarity and completeness of the prompts, the flexibility in handling recipe variations, and the analyzing the overall accuracy of the generated task trees in reflecting the intended cooking procedures. This comparative analysis will provide valuable insights into the effectiveness of different prompting techniques for task tree generation with LLMs.

The effectiveness of a particular prompting approach can depend on several factors, including the complexity of the recipe, the LLM's capabilities, and the desired level of detail in the generated task trees. Structured prompting might be suitable for simple recipes with well-defined steps. However, for more complex recipes with variations, open-ended or ingredient-centric prompting might offer more flexibility for Gemini to handle these variations. Additionally, the chosen approach should consider the LLM's capabilities. If the LLM struggles with reasoning or lacks sufficient knowledge about cooking procedures, a more structured approach with clear instructions might be necessary. Ultimately, the best prompting approach may involve a combination of techniques, tailored to the specific task and LLM being used.

IV. EXPERIMENTS / DISCUSSION

A. Experimenting on different prompting approaches

This study contained experiments with a head-to-head comparison of different prompting techniques for forming task trees with a large language model such as the Gemini. The three significant approaches examined were structured prompting, open-ended prompting, and zero-shot prompting. The performance evaluation demonstrated substantial discrepancies in the method's effectiveness and appropriateness of task tree formation. Among all the approaches, structured prompting is one of the most prospective, generating task trees in a more accurate and structured way. Through such structure, prescribing specific guidelines and standards makes the prompting more efficient, ensuring appropriate ingredients, steps, and needed structure. Hence, the task trees are reliable and coherent [1]. The structured nature of this approach was instrumental in eliminating disparities inherent in the outputs of large language models and, in this way, led to higher accuracy levels and consistency of results from different cooking activities [2].

Interestingly, open-ended prompting, which will provide freedom and space for individual considerations, failed to keep the pictures and similarity in the generated task structures. The absence of a clear structure resulted in different output formats with the contents. Thus, ensuring the precision and faithfulness of the created task trees became complicated [3]. Task tree creation with task-specific training examples was needed for complex cooking tasks. General language protocols and the lack of expertise led to the construction of flawed and imprecise task trees that were especially prevalent in tasks that demanded rules and convention understanding [4]. However, zero-shot prompting was identified to be a prospective baseline approach when task-specific data access is restricted and served as a jumping pad for further refinement and adaptation [5].

The advantages and disadvantages of several ways of promoting served as an instrument for a more profound understanding of their benefits and drawbacks. Using structured prompts allowed us to put guidelines for generating task trees with appropriate templates in place, thus achieving consistency and accuracy. It gave the robust support necessary for producing high-quality and accurate task trees, which were the preferred applications with many acceptable task outputs required [6]. Yet, the inflexibility of the structured approach may lower its effectiveness in revealing uniqueness, uncommonly used cooking methods, or something like that. Open-ended prompting, although enabling individual creativity and flexibility, presents a lack of consistency and structure and, therefore, needs to be more suitable for application in standardized task tree creation. Zero-shot prompting demonstrated the capability to generate task trees without needing training data specific to a particular task; however, this method failed when dealing with complex tasks and the deep knowledge required for accurate and trustable outputs [8].

The optimal prompting technique depends on the specific application and desired outcome. If high accuracy and detailed task trees are paramount, structured prompting might be the best choice. However, if user effort is a major concern and some level of flexibility is acceptable, open-ended prompting could be a viable option. Zero-shot prompting might be suitable for initial explorations where minimal setup time is crucial, but it may require further refinement for real-world robotic cooking tasks.

B. Discussion

Though task tree accuracy was the focus of our investigation, prompting strategies should be considered in future studies to encourage creativity and adaptation in robotic cookery. While structured prompting works well for reproducing recipes, it may prevent the robot from trying out new culinary combinations. Because open-ended prompting enables the LLM to use its own knowledge base to recommend unusual item pairings or cooking techniques, it may inspire more imaginative task trees. Moreover, zero-shot prompting in conjunction with reinforcement learning methods may allow the robot to modify its task trees in response to real-time sensor feedback or unforeseen circumstances that arise while cooking. Examining the ways in which various prompting methods affect these elements of robotic cooking may open new avenues for AI-driven culinary innovation.

C. Accuracy Results

The precision results from the experiments and peer review process were in-depth evidence that showed the effectiveness of separate prompting approaches. The most accurate scores were generally attained through structured prompting, with about 80% of the completed task trees by the reviewers being recognized as correct and complete. In every case, the accuracy of structured prompts showed the reliability and effectiveness of this method. The high scores of the structured prompting can be attributed to the effectiveness of its clear guidelines and templates, which enabled the model to output the necessary components, steps, and output organizations that were needed for proper execution.

Nevertheless, open-ended prompting and zero-shot prompting showed lower accuracy than structured prompting in this case. The lowest accuracy scores mostly fell in the range of 60-70%, suggesting that a non-trivial portion of the task trees contained incorrect information, missing elements, or errors. One reason for this low accuracy may be the absence of a stable structure and the use of individual feelings and capacities, which caused subjectivity and inconsistency of the task trees to be formed. The worst performer was the zero-shot prompting, and the accuracy metrics remained below 50% in most of the cases. The lack of task-specific training data for zero-shot prompting undermined its efficiency in creating task trees, which accurately reflected complex cooking procedures, especially those that needed domain expertise and wisdom.

D. Future work

The experiments and analysis conducted in this study provided valuable lessons and insights for future work in the field of task tree generation using large language models. The crucial thing to take away from this project is that proper prompts of a structured nature are very essential for accurate and reliable cooking task tree construction. The success of the structured prompting demonstrates the requirement of clear guidelines and templates that can be used to guide the language model in generating task trees, which should be designed in the required format and have all the necessary information [7]. The next phase of research needs to concentrate on enhancing and broadening the prompt system and exploring a way to combine more flexibility while at the same time preserving the good features of the structured output.

Another crucial lesson is how open-ended, and zero-shot prompts for task generation behave. The low accuracy scores and inconsistencies observed in these methods indicate the necessity of providing task-specific training data and special knowledge to get reliable and precise task trees. Future work should focus on improving the quality of generation by open-ended and zero-shot prompting, such as introducing domain-specific knowledge or employing transfer learning techniques to fine-tune the language model to the needs of a multi-level task tree [6]. Besides that, further study will examine the prospects for hybrid methods that integrate the strong points of both structured prompting and unstructured prompting, trying to reach the balance between accuracy and the ability to adapt to different circumstances.

As a rule, the experiments and analyses conducted in this study provides a lot of important information about the best prompting strategies to be utilized when using large language models in the development of task trees. Among the explored methods, the idea of structured prompting provided the most noticeable results, achieving the maximum precision and the slightest degree of error when developing task trees for cooking tasks. The principal points shown in this study underscore the need for structure rules, task-specific training data and joint work in building the field of task tree creation. In the future, emphasis should be on developing and perfecting the structured prompting techniques and mixed methods exploration as well as creating shared research platforms to organize activities aimed at enhancing the development of this ever-growing field.

REFERENCES

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

[2] Sakib, M. S., & Sun, Y. (2024). Consolidating Trees of Robotic Plans Generated Using Large Language Models to Improve Reliability. International Journal of Artificial Intelligence and Robotics Research, 2450002.

[3] Sakib, M. S., & Sun, Y. (2023). 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.

[4] Sakib, M. S., Paulius, D., & Sun, Y. (2022). Approximate Task Tree Retrieval in a Knowledge Network for Robotic Cooking. IEEE Robotics and Automation Letters, 7(4), 11492-11499.

[5] Paulius, D., Dong, K. S. P., & Sun, Y. (2021). Task planning with a weighted functional object-oriented network. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 3904-3910). IEEE.

[6] Sakib, M. S., Baez, H., Paulius, D., & Sun, Y. (2021). Evaluating recipes generated from functional object-oriented network. arXiv preprint arXiv:2106.00728.

[7] Sun, Y., Ren, S., & Lin, Y. (2014). Object–object interaction affordance learning. Robotics and Autonomous Systems, 62(4), 487-496.

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

[9] Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., ... & Garg, A. (2023). ProgPrompt: program generation for situated robot task planning using large language models. Autonomous Robots, 47(8), 999-1012.