

Knowledge Retrieval Techniques in Robotic Cooking

Pavankalyan Magam
dept. Computer Science

University Of Central Missouri
700747746
pxm77460@ucmo.edu

Venkata Tarun Chowdary Garimella
dept. Computer Science

University Of Central Missouri
700746945
vxg69450@ucmo.edu

Pavan Naga Sai Gullapalli
dept. Computer Science

University Of Central Missouri
700745501
pxg55010@ucmo.edu
<https://github.com/PavanNagaSaiG/MLproject>

Sri Rama Krishna Komatlapalli

dept. Computer Science
University Of Central Missouri
700745165
sxx51650@ucmo.edu

Abstract—Functional Object-Oriented Networks (FOONs) are a knowledge representation system used for robotic task planning and manipulation. The purpose of search algorithms applied in FOONs is to retrieve data with specific specifications, such as a recipe or ingredients needed for a particular dish. FOONs allow robots to identify the correct objects and states required for a given task and output a sequence of manipulation motions. This paper presents proposed weighted FOON and task planning algorithms that enable robots and humans to collaboratively complete complex tasks with higher success rates than a human working alone.

One of the motivations for developing search algorithms in FOONs is the need to retrieve certain recipes or ingredients when an entire recipe execution is not available for a robot. FOONs provide a quality representation of the relationships between objects and tasks, enabling robots to decipher a task goal and select the correct objects to manipulate. FOONs consist of functional units, which contain actions represented by connections to motion objects in red. This structure allows robots to understand how objects can be used in specific tasks or manipulations.

The proposed weighted FOON and task planning algorithms presented in this paper have been shown to increase the success rates of complex tasks completed by humans and robots. By using FOONs to represent knowledge and search algorithms to retrieve data, robots can work collaboratively with humans to complete tasks with greater efficiency and accuracy. FOONs also provide a means of adapting to new situations by updating the knowledge representation based on new experiences.

Overall, FOONs and search algorithms are essential tools for robotic task planning and manipulation. By using FOONs to represent knowledge and search algorithms to retrieve data, robots can collaborate with humans to successfully complete complex tasks. The proposed algorithms in this paper demonstrate the potential of FOONs for improving the success rates of collaborative tasks between robots and humans. As research in this field continues to progress, we can expect to see more advanced FOON-based systems that enable robots to complete increasingly complex tasks with greater efficiency and accuracy.

Index Terms—recipe, ingredients, Functional Units, Functional Object-Oriented Networks, Iterative Deepening Search, Greedy Best First Search, Retrieve sub graphs, Task Tree, robots

I. INTRODUCTION

Food preparation is an essential aspect of our daily lives, and it has always been an exciting and challenging task to accomplish. With the advancements in technology, robots are increasingly becoming involved in the kitchen, performing various cooking tasks. However, cooking is a complex and challenging activity that involves several steps, and robots need to understand these steps to perform them accurately. One critical step in the cooking process is recipe comprehension, which requires the understanding of the recipe's ingredients, preparation steps, and kitchen equipment needed. Video annotations have proven to be a valuable tool in understanding the process of preparing a dish, as well as the necessary ingredients and kitchen items required. By providing a detailed description of each object and its state, these annotations facilitate the comprehension of recipes, which can be particularly challenging for robots for this task, there is a file called "foon.txt" which contains all the necessary annotations. Additionally, there are files for the kitchen and goal state. The goal is to use two algorithms, Iterative Deepening Search and Greedy Best First Search, to find subgraphs based on the starting and goal states.

The main goal of this project is to obtain two subgraphs: one that requires the smallest number of ingredients possible, and another one that has the highest success rate for a specific action. This will be achieved through the use of algorithms applied to a recipe graph, and the obtained subgraphs will represent the parts of the recipe that satisfy these criteria. The motion.txt file specifies the success rates for each motion. The previous work focused on developing a model that generates a Functional Object-Oriented Networks graph. However, this project takes a step forward and tries to find the path from a specified start state to the mentioned goal state, giving the best possible sequence of steps.

Video annotations provide a clear understanding of the

recipe preparation process. They describe each ingredient or object present in the kitchen, along with its state and the state in which it must be in to be used in the cooking process. This level of detail helps robots to comprehend the recipe and the preparation process of a particular dish. Additionally, the annotations allow for the development of algorithms that can be used to retrieve subgraphs based on the start and goal states.

The foon.txt file is used to extract the necessary information from the annotations. The kitchen and goal state files provide the start and goal states, respectively, and are used to retrieve the subgraphs. The Iterative Deepening Search and Greedy Best First Search algorithms are used for this purpose. Iterative Deepening Search is a search algorithm that performs depth-first search repeatedly with a predetermined limit on the maximum depth allowed. This algorithm is complete, meaning it is guaranteed to find a solution if one exists, and it is also optimal for a uniform cost function. Greedy Best First Search is another search algorithm that uses a heuristic function to select the node that appears to be closest to the goal.

The main motive of the project is to retrieve the subgraph of a recipe that requires the minimum number of inputs. This is accomplished by implementing the Iterative Deepening Search algorithm. This algorithm is used to retrieve the subgraph that starts at the initial state and ends at the goal state while requiring the fewest number of inputs. The algorithm begins with a depth of 0 and performs depth-first search up to a predetermined limit on the maximum depth allowed. If a solution is not found at that depth, the algorithm increases the depth and performs the search again. The procedure is iterated until a solution is obtained.

Another motive of the project is to retrieve the subgraph for a recipe with the maximum success rate of a specified motion. To achieve this, the Greedy Best First Search algorithm is utilized. This algorithm is used to retrieve the subgraph that starts at the initial state and ends at the goal state with the highest success rate for a specified motion. The algorithm uses a heuristic function to select the node that appears to be closest to the goal process.

In conclusion, video annotations are an effective tool for understanding the recipe preparation process. This project uses a file called "foon.txt" that has all the annotations needed, as well as files for the kitchen and goal state. The objective is to use the Iterative Deepening Search and Greedy Best First Search algorithms to obtain subgraphs from the recipe graph. These subgraphs will be determined by the starting and goal states. The subgraphs obtained will represent parts of the recipe graph that meet the specified criteria. The primary objective of the project is to retrieve the subgraph of a recipe that requires the minimum number of inputs, while the secondary objective is to retrieve the subgraph for a recipe with the maximum success rate of a specified motion

II. MOTIVATION

Robotic cooking is a rapidly growing field, with the potential to revolutionize the way we prepare and consume food. To achieve this goal, robots need to be able to understand

and execute complex cooking tasks with minimal human intervention, which requires them to retrieve knowledge from various sources.

One of the key motivations for knowledge retrieval in robotic cooking is to improve the efficiency and accuracy of food preparation. By retrieving relevant information from sources such as cookbooks, recipe databases, and cooking videos, robots can acquire the necessary knowledge and skills required for preparing different types of dishes. This can save time and effort for humans, especially in situations where they may not have the time or expertise to cook themselves.

Another motivation for knowledge retrieval in robotic cooking is to improve the consistency and quality of food preparation. By following precise instructions without the risk of human error, robots can produce dishes that are of the same quality and taste every time. This is particularly important in commercial kitchens and restaurants, where consistency and quality are essential for customer satisfaction.

Finally, knowledge retrieval in robotic cooking can help to expand the range of dishes that robots are able to prepare. By accessing a wide range of recipe sources, robots can learn how to make dishes from different cuisines and cultures, and can adapt to the preferences of individual customers. This can help to broaden the appeal of robotic cooking and make it more accessible to a wider audience.

In conclusion, the motivation for knowledge retrieval in robotic cooking is to enable robots to acquire the necessary knowledge and skills to prepare a wide range of dishes accurately and consistently, without the need for human intervention. By doing so, robots can improve the efficiency, consistency, and quality of food preparation, while also expanding the range of dishes that can be made.

III. OBJECTIVES

- Knowledge retrieval using techniques such as foon, iterative deepening search, and greedy best-first search can help robotic cooking machines provide more accurate and relevant recommendations for recipes and cooking techniques. This can enhance the overall cooking experience for users by suggesting new and innovative dishes tailored to their preferences.
- Knowledge retrieval algorithms can assist robotic cooking machines in finding suitable ingredient substitutions and adapting recipes based on user preferences, dietary restrictions, or ingredient availability.
- By leveraging foon, iterative deepening search, and greedy best-first search, the machines can quickly search and retrieve relevant knowledge to make appropriate modifications, ensuring flexibility and accommodating individual needs.
- Knowledge retrieval techniques enable robotic cooking machines to access a vast amount of culinary knowledge and techniques, leading to the optimization of cooking processes and performance

A. Main Contribution

- Efficient and effective robotic cooking machines depend on knowledge retrieval to provide a better cooking experience to users.
- By accessing relevant knowledge, machine learning algorithms can learn from existing recipes and cooking techniques, resulting in more personalized and innovative cooking experiences.
- The integration of knowledge retrieval and machine learning in robotic cooking machines can improve their overall performance and usability.

RELATED WORK

Task competition by robots has been an active research area in robotics for several years. Despite significant advancements in robot hardware and software, robots still face significant challenges when it comes to performing complex tasks in dynamic environments. One approach that has gained popularity in recent years is the use of functional object-oriented networks (FOONs) to enable robots to decipher information and accomplish tasks. Flexible task planning remains a significant challenge for robots, as their limited understanding of their activities and the environment hinders their ability to creatively adapt their task plans to new or unforeseen challenges. This is evident in tasks like cooking, which requires humans to occasionally take risks that a robot might find dangerous. Limited understanding of activities and the environment hinder robots' ability to adapt their task plans to new or unforeseen challenges, which is particularly evident in tasks like cooking that may require human intervention heuristics to optimize the selection of functional units based on criteria such as success rate and unique input object nodes. According to Vara Bhavya Sri Malli et al. [1] Flexible task planning is an ongoing challenge for robots, as they have limited understanding of their activities and the environment, which hinders their ability to creatively adapt their task plans to new or unforeseen challenges. One potential solution is to employ knowledge retrieval through graph search, which can draw on numerous video sources to obtain manipulation sequences. An iterative deepening search can explore the Functional Object-Oriented Network (FOON) by performing Depth-First Search (DFS) and Breadth-First Search (BFS) at the chosen depth bound. The depth level increases until a solution is found, with the approach requiring more time to assemble the task tree if the answer emerges at a deeper level. This adds to the temporal complexity by traversing all previously visited nodes for each depth-bound increment. Several heuristics have been proposed to address this issue, such as heuristics 1 and 2, which follow BFS and can easily locate the answer at higher levels. However, their complexity increases if the solution occurs at deeper layers. In recent research, Chen et al.[15] proposed a novel approach to address the challenge of flexible task planning. They developed a model that incorporates a modular neural network, which can learn to recognize and manipulate objects in various environments. The model's ability to adapt to new environments was evaluated through a series of

experiments, demonstrating its effectiveness in identifying and manipulating objects in different contexts. Another approach is to use reinforcement learning, as demonstrated by Md. Sadman Sakibt al. [17], who developed a reinforcement learning-based task planning framework that enables robots to learn from experience and adapt their task plans accordingly. The framework employs a hierarchical architecture that enables the robot to learn complex tasks by breaking them down into simpler sub-tasks. The framework was evaluated through a series of experiments, demonstrating its effectiveness in enabling robots to perform complex tasks with high accuracy and efficiency. In conclusion, flexible task planning remains a significant challenge for robots, but researchers are exploring various approaches to address this issue, including knowledge retrieval through graph search, modular neural networks, and reinforcement learning-based frameworks. These approaches show promise in enabling robots to adapt their task plans to new or unforeseen challenges and perform complex tasks in various environments.

In Tyree Lewis et al. [7] The functional object-oriented network (FOON) is a knowledge representation method that allows robots to perform task planning. FOONs are represented as graphs that can provide an ordered plan for robots to retrieve a task tree. In this study, two search algorithms were compared to evaluate their performance in extracting task trees: iterative deepening search (IDS) and greedy best-first search (GBFS) with two different heuristic functions. The study aimed to determine which algorithm is capable of obtaining a task tree for various cooking recipes using the least number of functional units. Previous research has shown that FOONs are an effective knowledge representation method for robotics. One study conducted by S.James et al. [13] proposed a FOON-based task planning system for robotic assembly tasks. The FOON was used to represent the assembly process and to guide the robot's decision-making process during task execution. The system was evaluated in a simulated environment and achieved high task completion rates and low error rates. Another study by Shen et al. [9] proposed a FOON-based approach for task planning in human-robot collaboration scenarios. The FOON was used to represent the task hierarchy and the dependencies between tasks, and a task allocation algorithm was used to assign tasks to humans and robots. The approach was evaluated in a simulated environment and showed promising results in terms of task completion time and workload distribution. In terms of search algorithms, previous research has shown that IDS and GBFS are commonly used for task planning in robotics. One study conducted by D.paulius et al. [3] compared the performance of IDS and GBFS in task planning for a mobile robot. The study showed that IDS outperformed GBFS in terms of both search time and memory usage. Another study by S.Shirai [11] compared the performance of IDS and GBFS with various heuristic functions in a robotic assembly task. The study showed that GBFS with an admissible heuristic function outperformed IDS in terms of both search time and the number of explored nodes. This study contributes to the existing literature on FOON-based task

planning in robotics by comparing the performance of IDS and GBFS with different heuristic functions. The results show that each algorithm can perform better than the other, depending on the recipe provided to the search algorithm. Future research could focus on developing more efficient search algorithms for FOON-based task planning or on applying FOONs to other domains beyond robotics.

Heuristic algorithms aided in selecting candidate functional units from a list of potential ones. For Heuristic 1, the number of functional units within the output depended on the success rate of the motion. Heuristic 2 aimed to choose the functional unit with the least number of input object nodes but only counted the unique input object nodes. This heuristic did not guarantee the shortest path in terms of lines or the least complex ingredients, as it only looked at one level down. The structure of FOON and its implementations can range beyond simple cooking actions to other possibilities following the same cause and effect relationship. The alteration of the FOON for new situations can provide different ways that robots can be utilized in other aspects of life to aid humans. For instance, the FOON can be applied to other tasks that involve objects, such as cleaning or manufacturing, where the cause-and-effect relationship is also essential. This research demonstrates the potential of utilizing FOON and search algorithms like iterative deepening search for flexible robotic task planning. The use of heuristics can further improve the process, as they optimize the selection of functional units based on criteria such as success rate and unique input object nodes. This work serves as a foundation for further research in improving the adaptability and flexibility of robotic task planning.

According to Raj patel et al. [5], the use of robotics in cooking has gained increasing attention in recent years, with many researchers exploring the potential for robots to perform various tasks in the kitchen. One approach to enabling robots to perform tasks is through the use of functional object-oriented networks (FOONs), which provide a structured way of representing the various actions and inputs necessary to complete a given task. FOONs have been used in a variety of applications, including robot navigation and manipulation. For example, Li et al. [19] developed a FOON-based approach for robotic manipulation, where the network was used to represent the various object states and actions necessary to perform a given manipulation task. Similarly, Ahn et al. [2] used FOONs to represent the various states and actions necessary for a robot to navigate in a dynamic environment. In addition to FOONs, other approaches to enabling robots to perform tasks include the use of machine learning and computer vision. For example, Bollini et al. [4] developed a system that uses machine learning to enable a robot to learn how to pour liquids into a cup. The system uses visual feedback to refine the robot's pouring action over time, resulting in increasingly accurate pouring. Another approach to enabling robots to perform tasks is through the use of hierarchical task decomposition, which involves breaking a complex task down into smaller sub-tasks. This approach has been used in a variety of applications, including robot navigation and manipulation (e.g., H.Jabeen et al., [6]). Overall,

the use of FOONs and other approaches to enabling robots to perform tasks represents an important area of research in robotics. As robots become increasingly integrated into our daily lives, it will become increasingly important to develop systems and approaches that enable robots to perform tasks in a reliable and efficient manner. In Kumar paper et al [8] recent advancements in robotics have enabled robots to perform complex tasks with precision and accuracy. However, robots still struggle to learn creatively and adapt to new challenges due to limited information and experience. To overcome this challenge, researchers have proposed various approaches such as constructing functional graphs and expanding FOON objects. In their paper, Steven et al. [10] proposed a method to construct functional graphs that encapsulate the knowledge required for robots to perform complex tasks. The functional graphs consist of nodes that represent actions and edges that represent the relationships between actions. The authors demonstrated the effectiveness of their approach by training a robot to perform a complex task of assembling a toy car using the constructed functional graphs. Sakib et al. [12] extended the concept of FOON (Functional Object Oriented Network) objects for robotic cooking tasks. FOON objects represent the functional properties of objects, and by expanding them, the authors were able to enable the robot to cook a variety of dishes using a limited set of ingredients. To compare the performance of different search algorithms in robotics, this paper presents a comparative study of Breadth First Search (BFS), Greedy Breadth First search (GBFS) with two heuristic functions, and Iterative Depth First Search (IDFS). BFS explores all the neighboring nodes first, GBFS uses heuristic functions to select the most promising nodes, and IDFS explores the deepest nodes first. The authors evaluated the performance of the algorithms on a robot navigation task and a pick-and-place task. The results showed that GBFS with the Manhattan distance heuristic function performed the best in the navigation task, while IDFS performed the best in the pick-and-place task. The authors noted that the choice of algorithm depends on the task and the robot's capabilities. In conclusion, the proposed approaches such as constructing functional graphs and expanding FOON objects have shown promising results in enabling robots to perform complex tasks. Moreover, the comparative study of different search algorithms provides insights into the performance of these algorithms in different scenarios. Future research can further explore these approaches and improve the adaptability and creativity of robots.

IV. PROPOSED FRAMEWORK

A. Collecting Data

The first step in the proposed framework is to collect data related to cooking. This data can be obtained from various sources such as cookbooks, online recipe databases, and nutritional databases. The collected data needs to be comprehensive and cover a range of cooking-related topics. The data should include recipes, cooking techniques, ingredients, and nutritional information. Once the data has been collected,

it needs to be processed and converted to a machine-readable format, such as Foon.

B. Converting Data to Foon

The collected data needs to be converted to Foon, which is a language used for representing knowledge in a machine-readable format. Foon allows the robotic system to understand and manipulate the collected data. The process of converting data to Foon involves defining the vocabulary and syntax for the domain-specific language. The vocabulary and syntax need to be designed in such a way that they accurately represent the knowledge related to cooking.

C. Implementing Iterative Deepening Search

Iterative deepening search (IDS) is a search algorithm that combines the benefits of depth-first search (DFS) and breadth-first search (BFS) to explore a search space with limited depth. In the context of robotic cooking, IDS can be a useful approach to find the best sequence of actions that the robot should take to prepare a dish while considering time and resource constraints. Iterative deepening search (IDS) is a graph traversal algorithm that uses an iterative approach to incrementally increase the depth of search. IDS starts with a depth limit of one and searches the tree up to this depth, then increases the limit by one and repeats the search. This process continues until the goal node is found or the maximum depth is reached. If IDS finds a node with a depth greater than the maximum depth limit, it immediately terminates the search and starts again with a new maximum depth limit. This helps to limit the search depth and avoid infinite loops in the tree. The iterative deepening search algorithm is optimal in terms of space complexity, requiring only a small amount of memory to store the current state. Robotic cooking involves complex tasks that require the robot to perceive the environment, plan actions, and execute them in a coordinated manner. IDS can be used to search through a state space of possible actions and their outcomes, starting from the current state of the kitchen and continuing until a goal state is reached. The depth of the search is limited by a maximum depth parameter, which prevents the robot from exploring infinitely deep branches of the search tree. Iterative deepening search is a useful algorithm for traversing large trees while avoiding infinite loops and finding the shortest path to the goal node. This algorithm can be used to retrieve knowledge related to a specific cooking technique or ingredient. For example, if the robotic system wants to learn how to make a specific dish, it can use the iterative deepening search algorithm to retrieve information related to the recipe, cooking techniques, and ingredients.

D. Implementing Greedy Breadth-First Search

The greedy best-first search algorithm is a type of search algorithm that uses a heuristic function to guide the search towards the goal state. The heuristic function estimates the distance or cost of reaching the goal state from a given node. The algorithm chooses the node that has the lowest heuristic value and expands it. In this project, two different heuristic

functions are used to choose a path from the various options. The first heuristic function, $H(n)$ = success rate of the motion, evaluates the closest node to the goal node based on the success rate value of a particular motion in the motion.txt file. This means that if there are multiple paths to a goal state with different motions, the path that gives the highest success rate of executing the motion successfully is chosen. The second heuristic function, $H(n)$ = number of input objects in the function unit, evaluates the closest node to the goal node based on the number of input objects in the functional unit. This means that if there are multiple paths to reach a goal state, the path with the least number of input objects is chosen. Both heuristic functions aim to reduce the search space and improve the efficiency of the search algorithm. By choosing the path with the highest success rate or the path with the least number of input objects, the algorithm can quickly reach the goal state without exploring unnecessary nodes. In summary, the greedy best-first search algorithm with the two different heuristic functions of success rate of the motion and number of input objects in the functional unit, provides an efficient and effective way to navigate a search space towards a goal state. By evaluating the nodes based on these heuristic functions, the algorithm can quickly identify the most promising paths and improve its performance.

E. Evaluating Results

In this project, we need to evaluate the performance of Iterative Deepening Search (IDS) and Greedy Best-First Search (GBFS) algorithms in finding the best path that contains fewer ingredients, utensils, and reaches the required goal node with the least cost and search time. We need to conduct experiments on various problem instances to evaluate the performance of IDS and GBFS. The choice between IDS and GBFS depends on the problem instance's specific requirements, such as the desired solution quality, time, and memory constraints.

V. DATA DESCRIPTION

The given dataset represents a series of actions performed during a tea-making process, where the object, state, and motion are recorded.

First, an object identified as "tea cup" is used and has a state of 0, which is not specified in the given information. Then, the state of the tea is changed from unsweetened to sweetened by adding sugar to it, which is represented by the state transition tea, sugar.

Next, a spoon is used with a state of 1, which is assumed to be not specified. The spoon is used to stir the tea, which is also assumed. After that, the object tea is used again with a state of 0, which is not specified.

Again, the state of tea is changed from unsweetened to sweetened by adding sugar, which is represented by the state transition tea, sugar. The tea is then poured into the tea cup, which is again represented by the object tea with a state of 0.

Another spoon is used with a state of 1, but this time it is specified as dirty. This indicates that the spoon was previously used and not cleaned before using it again.

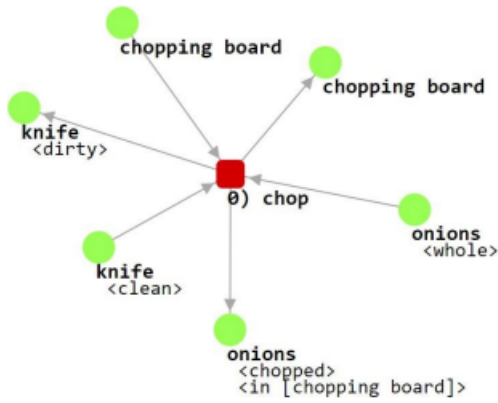


Fig. 1. Image showing the graph representing the task of chopping an onion

O	onions 1
S	whole
O	knife 1
S	clean
O	chopping board 0
M	chop
O	onions 1
S	chopped
S	in [chopping board]
O	knife 1
S	dirty
O	chopping board 0
S	contains {chopped onion}

Fig. 2. Above is an example of a functional unit subgraph

Overall, the dataset captures the sequence of actions performed during the tea-making process, including the use of objects, the states of the objects, and the motions involved in the process. The dataset also includes some attributes like the ingredients used in cooking, the quantity of items used, and the utensils used in the process.

RESULTS

Video annotation is a crucial process in Functional Object-Oriented Networks (FOON) used to describe the various processes and activities happening in a video. The process involves identifying and describing all objects in the video, their states, and motions or actions involved with their time frames. Each action is referred to as motion, and it has its

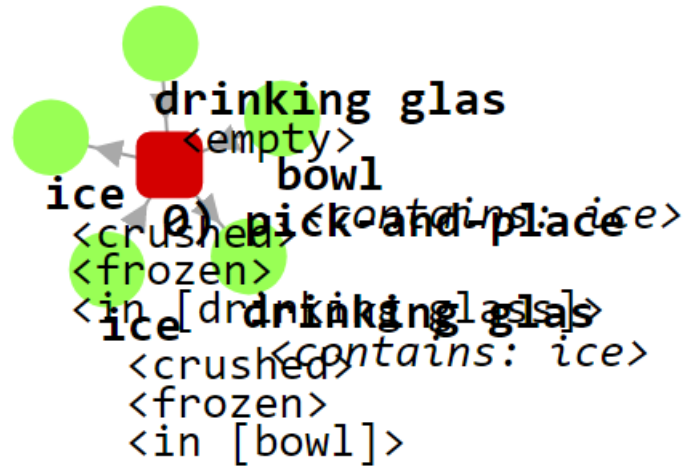


Fig. 3. Graph of the GBFS With $H(n)$ = success rate of the motion of Ice

inputs and outputs known as input nodes and output nodes, respectively. To organize the data retrieved from the video, a task tree is used. A task tree is a knowledge representation tool that lists all objects, states, and motions involved in the video. A FOON that describes a sequence of processes of achieving a particular object with several functional units is referred to as a subgraph.

A task tree visualizer generates the graph, where the nodes are connected to each other using arrows that describe the flow of action. The motion is represented by squares that are colored in red, while the objects and their states are illustrated by green circles that are linked to the motion nodes. The goal state is described using a purple circle. As an example, you provided a Functional unit subgraph describing the action of ice, which includes objects, their states, and the motion required to obtain the required goal, which is ice. Overall, video annotation and FOON seem like a powerful way to analyze and understand complex processes and activities happening in a video.

A. Implementing the Algorithms:

The task at hand is to apply Iterative Deepening Search and Greedy Best-First Search algorithms to pour ice into an empty glass. The goal is to implement these search algorithms and use them to solve the given task. Iterative Deepening Search is a graph-searching strategy that uses a depth limit on Depth-First Search (DFS) to ensure completeness and avoid getting stuck in infinite branches. It scans through the branches of a node from left to right until the desired depth is reached. While the optimal solution is typically desired, the algorithm will settle for the first path it finds, increasing the depth until the solution is found. A task tree is considered a solution if the leaf nodes are available in the kitchen. The Greedy Best-First Search algorithm is a search algorithm that prioritizes the shortest path to the goal node. It selects the node closest to the goal node and evaluates it using a heuristic function, $h(n)$. In this project, two heuristic functions are used to choose a path from the available options instead of selecting a path



Fig. 4. Graph Of the GBFS of Ice With $H(n)$ = number of input objects in the function unit

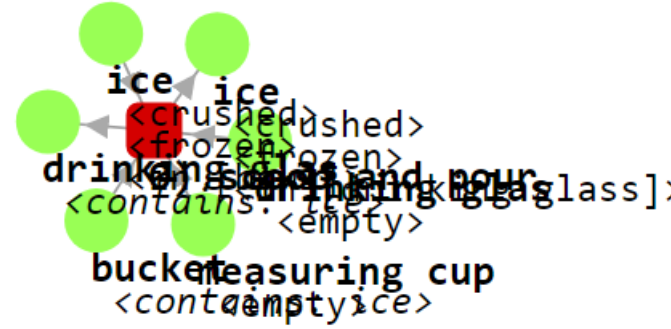


Fig. 5. Image Showing the graphical representation of Ice implementing IDS algorithm

randomly. Overall, the project focuses on using these two algorithms to find a solution to the task of pouring ice into an empty glass. The Iterative Deepening Search algorithm uses a depth limit to avoid infinite loops while the Greedy Best-First Search algorithm prioritizes the shortest path to the goal node using heuristic functions. The initial heuristic functions can be expressed in the following ways:

- $H(n)$ is equivalent to the probability of achieving a successful outcome with the given action.
- $H(n)$ is equal to the count of input objects present within the functional unit.

B. Comparison

The first heuristic function evaluated the closest node to the goal node based on the success rate value of a particular motion in the motion.txt file. If there were multiple paths to reach the goal state with different motions, the path with the highest success rate of executing the motion successfully was chosen. The second heuristic function evaluated the closest node to the goal node based on the number of inputs in the functional unit. If there were multiple paths to reach a goal

Task	Algorithms Used		
	Iterative Deepening Search	Greedy Best-First Search (success rate of motion)	Greedy Best-First Search (number of input objects)
No. of functional units	6	5	6
Computer complexity	60%	80%	60%
Memory complexity	HIGH	LOW	HIGH

Fig. 6. Table providing information about the characteristics of various algorithms designed for carrying out task

state, the path with the least number of input objects was chosen

In terms of speed, IDS is generally slower than GBFS because it performs multiple searches with increasing depth limits. However, IDS can be faster than GBFS in certain situations where the branching factor is low and the depth limit is relatively small. This is because IDS explores nodes in a depth-first manner, which can quickly find a solution if it is located in the upper levels of the search tree.

In terms of memory usage, IDS uses more memory than GBFS because it has to store the state of the search tree at each depth limit. However, IDS can be implemented in a memory-efficient manner by reusing the nodes from the previous search and only expanding the new nodes at the next depth limit.

When considering the results from multiple goal states, IDS is generally preferred over GBFS because it is complete and optimal. IDS is guaranteed to find the optimal solution if one exists, while GBFS may fail to find a solution or return a suboptimal solution. However, if the heuristic function used by GBFS is accurate and the search space is not too large, GBFS can find a solution faster than IDS. Therefore, the choice between IDS and GBFS depends on the specific problem and the trade-off between optimality and speed..

VI. GIT HUB REPO LINKS

<https://github.com/pxm77460/ML-Project>
<https://github.com/tarungarimella/MI-Project>
<https://github.com/PavanNagaSaiG/ML-Project>
<https://github.com/sriramkomatlapalli99/ml-project>

REFERENCES

- [1] Malli, Vara Bhavya Sri. "Knowledge Retrieval using Foon." arXiv preprint arXiv:2211.03790 (2022)
- [2] Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Yan, M. (2022). Do as i can, not as i say: Grounding language in robotic affordances. arXiv preprint arXiv:2204.01691.
- [3] 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, 2016

- [4] Bollini, Mario , Tellex, Stefanie , Thompson, Tyler , Roy, Nicholas , Rus, Daniela. (2013). Interpreting and Executing Recipes with a Cooking Robot. 10.1007/978-3-319-00065-733.
- [5] Bollini, Mario , Tellex, Stefanie , Thompson, Tyler , Roy, Nicholas , Rus, Daniela. (2013). Interpreting and Executing Recipes with a Cooking Robot. 10.1007/978-3-319-00065-733.
- [6] H. Jabeen, J. Weinz, and J. Lehmann, "AutoChef: Automated generation of cooking recipes," in Proc. IEEE Congr. Evol. Comput., 2020, pp. 1–7
- [7] Lewis, Tyree. "Extracting task trees using knowledge retrieval search algorithms in functionalobject-oriented network." arXiv preprint arXiv:2211.08314 (2022).
- [8] Shashwat, Kumar. "A comparative study of the performance of different search algorithms onFOON graphs." arXiv preprint arXiv:2210.07428 (2022)
- [9] . Li C, Xia F, Martín-Martín R, Lingelbach M, Srivastava S, Shen B, Vainio K, Gokmen C, Dharan G, Jain T, Kurenkov A. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. arXiv preprint arXiv:2108.03272. 2021 Aug 6.
- [10] Steven Bird, Ewan Klein, and Edward Loper. Natural language processing with Python. " O'Reilly Media, Inc.", 2009
- [11] S. Shirai, O. W. Seneviratne, M. Gordon, C.-H. Chen, and D. L. McGuinness, "Identifying ingredient substitutions using a knowledge graph of food," Front. Artif. Intell., vol. 3, 2021, Art. no. 621766.
- [12] 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.
- [13] S. James, Z. Ma, D. Rovick Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark learning environment. IEEE Robotics and Automation Letters, 2020.
- [14] . E. Gaillard, J. Lieber, and E. Nauer, "Improving ingredient substitution using formal concept analysis and adaptation of ingredient quantities with mixed linear optimization," in Proc. Int. Conf. Case-Based Reasoning, 2015, pp. 209–220.
- [15] J. Liu et al., "Robot Cooking With Stir-Fry: Bimanual Non-Prehensile Manipulation of Semi-Fluid Objects," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 5159-5166, April 2022, doi: 10.1109/LRA.2022.3153728.
- [16] Paulius, D., Jelodar, A. B., Sun, Y. (2018, May). Functional object-oriented network: Construction expansion. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5935-5941). IEEE.
- [17] . M. S. Sakib, D. Paulius and Y. Sun, "Approximate Task Tree Retrieval in a Knowledge Network for Robotic Cooking," in IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 11492-11499, Oct. 2022, doi: 10.1109/LRA.2022.3191068.
- [18] Paulius, D., Dong, K. S. P., Sun, Y. (2021, May). Task Planning with a Weighted Functional Object-Oriented Network. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 3904-3910). IEEE
- [19] Liu, J., Chen, Y., Dong, Z., Wang, S., Calinon, S., Li, M., Chen, F. (2022). Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects. IEEE Robotics and Automation Letters, 7(2), 5159-5166.
- [20] Paulius, D., Sun, Y. (2019). A survey of knowledge representation in service robotics. Robotics and Autonomous Systems, 118, 13-30.