LLMBot: Multi-Agent Robotic Systems for Adaptive Task Execution

Harith Ibrahim, Shane Xi, Harith Alsafi

School of Computing, University of Leeds, Street, Leeds, LS2 9JT, West Yorkshire, United Kingdom.

Contributing authors: harithsami01@gmail.com; S.Q.Xie@leeds.ac.uk; harith.alsafi@gmail.com;

Abstract

Robotic task planning systems often require significant computational resources and financial investment, limiting their accessibility and scalability. Can we leverage recent advancements in Large Language Models (LLMs) to create more cost-effective and lightweight robotic planning solutions? This paper introduces an innovative framework that integrates LLMs with robotic systems for natural language interaction and task execution. Our approach, which we call LLMRP (Large Language Model Robotic Planner), combines GPT-3.5 Turbo for natural language understanding, reasoning, and high-level task planning with a simulated 3D environment in Unreal Engine. LLMRP uses behavior trees for robust low-level execution of robotic actions, grounding the LLM's capabilities in a practical robotic context. The system employs prompt engineering, multimodal input, and parameter optimization to enhance the LLM's performance and reduce computational overhead. For systematic evaluations, we developed a comprehensive test suite assessing success rates, spatial distributions, and cost-effectiveness across various scenarios. Experimental results demonstrate that our approach outperforms traditional planning systems in terms of computational efficiency and cost-effectiveness while maintaining comparable task success rates. Additionally, we have demonstrated LLMRP using a simulated mobile manipulator in Unreal Engine. This research advances multi-agent collaboration and human-robot interaction by enabling more intuitive and resource-efficient communication between humans and robots. By combining modern language models with lightweight execution frameworks, this project paves the way for more accessible, adaptive, and cost-effective robotic solutions capable of natural language-based collaboration.

Keywords: Large Language Models, Robotic Systems, Natural Language Interaction, Task Planning, Human-Robot Interaction, Multi-agent Collaboration, Warehouse

Automation, Behavior Trees, Simulated Environment, Physical Task Execution, Embodiment

1 Introduction

Effective robotic task planning requires not only a deep understanding of the environment and available actions but also the ability to interpret and execute high-level, natural language instructions. Traditional approaches to robot planning often involve complex, expensive algorithms that are difficult to develop and maintain. Can we leverage recent advancements in large language models (LLMs) to create more accessible and cost-effective robot planning solutions? Our work introduces LLM-RP (Large Language Model Robotic Planner), a novel framework that integrates LLMs with symbolic planning methods for intuitive, natural language-based robot control.

LLMs trained on vast corpora of text have demonstrated remarkable multi-task generalization and common-sense reasoning capabilities. Recent research has explored their potential in robotic task planning by generating or scoring action sequences. However, these approaches often lack grounding in the robot's physical capabilities and current environmental state. LLM-RP addresses this limitation by combining the rich language understanding of LLMs with the efficiency and adaptability of symbolic planners, specifically behavior trees. A key component of LLM-RP is its use of

Fig. 1 Diagram Showing Simplified Project Plan

GPT-3.5 Turbo for natural language understanding, reasoning, and high-level task planning. We leverage the fact that LLMs are trained on diverse web content, including programming tutorials and documentation, to create a novel prompting scheme. This scheme provides the LLM with a Pythonic-style program structure, including available actions, environment objects, and function definitions for tasks. We incorporate situated awareness by asserting preconditions and implementing recovery actions for failed assertions.

LLM-RP operates within a simulated 3D environment in Unreal Engine, allowing for comprehensive testing and evaluation. The system utilizes behavior trees for robust low-level execution of robotic actions, bridging the gap between high-level language understanding and concrete robot behaviors. We employ prompt engineering, multimodal input, and parameter optimization to enhance the LLM's performance and reduce computational overhead.

Our approach goes beyond simple language-to-action mapping. LLM-RP enables robots to interpret complex, free-form natural language instructions, reason about the given task, and coherently execute the necessary actions to accomplish the objective. Moreover, it exhibits adaptability to handle novel situations without extensive retraining or reprogramming.

To evaluate LLM-RP, we developed a comprehensive test suite assessing success rates, spatial distributions, and cost-effectiveness across various scenarios. Our results demonstrate that LLM-RP outperforms traditional planning systems in terms of computational efficiency and cost-effectiveness while maintaining comparable task success rates. We also showcase LLM-RP's capabilities using a simulated mobile manipulator in Unreal Engine, highlighting its potential for real-world applications.

This research advances multi-agent collaboration and human-robot interaction by enabling more intuitive and resource-efficient communication between humans and robots. By combining modern language models with lightweight execution frameworks, LLM-RP paves the way for more accessible, adaptive, and cost-effective robotic solutions capable of natural language-based collaboration.

2 Background and related work

2.1 Task Planning in Robotics

Traditional approaches to robotic task planning often involve search algorithms within pre-defined domains (Fikes and Nilsson, 1971; Jiang et al., 2018; Garrett et al., 2020). These methods can be computationally expensive and challenging to scale in environments with numerous feasible actions and objects (Puig et al., 2018; Shridhar et al., 2020). To address this, researchers have explored various techniques, including heuristic-guided search (Baier et al., 2007; Hoffmann, 2001; Helmert, 2006) and learning-based task and motion planning (Akakzia et al., 2021; Eysenbach et al., 2019; Jiang et al., 2019).

Our approach, LLM-RP, takes a different route by leveraging large language models to directly generate plans that include conditional reasoning and error correction, bypassing the need for extensive search.

2.2 Large Language Models and Conversational Tuning

Large Language Models (LLMs) like GPT-3 are neural networks trained on vast amounts of text data, enabling them to generate human-like text outputs. Their training process exposes them to a wide range of knowledge and scenarios, allowing them to exhibit reasoning, common sense, and goal-oriented behaviors. Recent work has explored the use of LLMs in robotic task planning (Ahn et al., 2022; Huang et al., 2022a, b; Li et al., 2022), but challenges remain in bridging the gap between language understanding and executable robot actions.

LLM-RP builds upon these advances by integrating GPT-3.5 Turbo for natural language understanding and high-level planning, while addressing the grounding problem through a novel prompting scheme and integration with symbolic planners.

2.3 Embodied Large Language Models

Recent research has explored techniques to enhance LLMs' capabilities for embodied environments like robotics, including prompt engineering, multimodal prompting, and few-shot learning. These techniques leverage language to define the rules of the world and the robot's abilities, a process called "grounding" (Huang et al., 2022b; Ahn et al., 2022).

Our work extends this concept by introducing a programming language-inspired prompting scheme that informs the LLM of both the situated environment state and available robot actions, ensuring output compatibility with robot capabilities.

2.4 Symbolic Planning Methods

Symbolic planning techniques, such as behavior trees, offer a simple yet powerful way to create adaptable logic and decision-making without the need for complex machine learning models. While recent works have explored learning-based approaches to task planning (Mirchandani et al., 2021; Nair and Finn, 2020; Shah et al., 2022), these methods often struggle with generalization to novel situations.

LLM-RP combines behavior trees with LLMs, leveraging the strengths of both components: LLMs provide rich language understanding capabilities, while symbolic planners offer computational efficiency and the ability to rapidly re-plan lower-level actions based on changes to the symbolic world state.

2.5 Cost-Effective Robotic Planning

A key challenge in robotic task planning is the development of sophisticated algorithms that are both effective and cost-efficient. Traditional approaches often require significant computational resources and financial investment, limiting their accessibility and scalability.

Our work addresses this challenge by proposing a lightweight, LLM-integrated approach that reduces computational overhead while maintaining high task success rates. By leveraging the pre-trained knowledge of LLMs and combining it with efficient symbolic planning methods, LLM-RP aims to provide a more accessible and cost-effective solution for robotic task planning.

2.6 Simulated Environments for Robotic Research

Simulated environments play a crucial role in robotics research, allowing for rapid prototyping, testing, and evaluation of algorithms without the need for physical hardware. Recent works have utilized game engines and physics simulators for robotic task planning and learning (Shah et al., 2018; Savva et al., 2019).

LLM-RP operates within a simulated 3D environment in Unreal Engine, enabling comprehensive testing and evaluation of our approach across diverse scenarios. This setup allows us to assess the system's performance, running costs, and cost-effectiveness in a controlled yet realistic setting.

3 Method

3.1 Architecture

The proposed system architecture integrates a Large Language Model (LLM) with robotic agents operating in a virtual Unreal Engine environment. The Python environment handles the LLM's logic, reasoning capabilities, and communication with the simulation via an API. Unreal Engine provides a realistic virtual testbed, leveraging its strengths in intelligent agent creation, behavior simulation, navigation, and perception systems.

Fig. 2 High level diagram of system architecture.

Simulation Environment Unreal Engine was chosen as the development platform due to its superior performance, visual quality, and powerful tools for defining and managing intelligent agents' decision-making processes and actions, including its visual scripting system, Blueprints, and its Behavior Tree system.

Fig. 3 Screenshot showing Robot Chat User Interface.

Fig. 4 Screenshot of the simulation. Intelligent robots managing inventory in a Warehouse.

Fig. 5 Warehouse Simulation Environment in Unreal Engine 5.

Algorithm 1 Calculate $y = x^n$

Require: $n \geq 0 \vee x \neq 0$
Ensure: $y = x^n$

$y \Leftarrow 1$
if $n < 0$ **then**
    $X \Leftarrow 1/x$
    $N \Leftarrow -n$
else
    $X \Leftarrow x$
    $N \Leftarrow n$
end if
while $N \neq 0$ **do**
    if N is even **then**
        $X \Leftarrow X \times X$
        $N \Leftarrow N/2$
    else[N is odd]
        $y \Leftarrow y \times X$
        $N \Leftarrow N - 1$
    end if
end while

3.2 Language Models and Behaviour Trees

The system leveraged GPT-3.5 Turbo, a state-of-the-art LLM, for natural language understanding, reasoning, and high-level task planning. Prompting techniques, such as zero and few-shot learning, chain of thought reasoning, and multimodal prompting, were employed to improve the model's performance and grounding in the current environment context.

Behavior trees were used to model and implement decision-making processes in the robotic agents. They offer a hierarchical and modular structure for representing complex behaviors, making them easy to design, debug, and maintain.

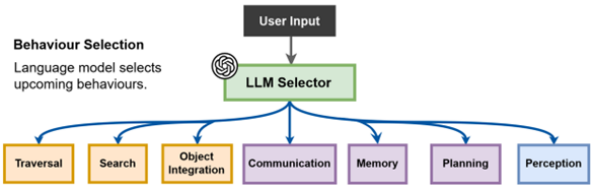

Fig. 6 Proposed approach of utilizing Language Models in Behaviour Trees.

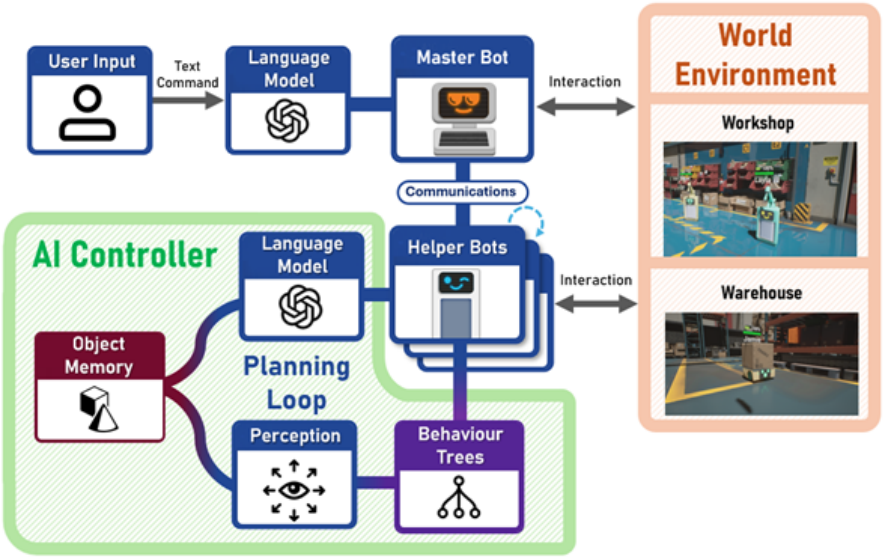

Fig. 7 Proposed Robot Cognitive Architecture.

3.3 Intelligent System Implementation: Robot Abilities and Communication

The robotic agents, called HelperBots, could navigate the virtual environment, locate and interact with objects, and perform various tasks based on natural language instructions. They used five high-level behaviors: "Go to" for navigation, "Object interact" for manipulating objects, "Refer to memory" for storing and retrieving environmental information, "Status" for reporting internal status, and "Communicate" for inter-agent communication.

Perception and Navigation For perception, the system assumed perfect object identification within the robot's line of vision. Regarding navigation, the system utilized Unreal Engine's AI navigation system, employing a technique called Navigation Mesh (NavMesh) for efficient pathfinding in the 3D environment.

Robot Memory and Hierarchical Architecture The system employed context-aware memory within the language model and object memory on the robot logic side. A "Master Robot" served as a central control unit, with access to the combined memory of all other robots. Several specialized robots, such as DronBot, ArmBot, FloorBot, and ShelfBot, collaborated in a simulated warehouse environment, with the Master Robot coordinating their efforts.

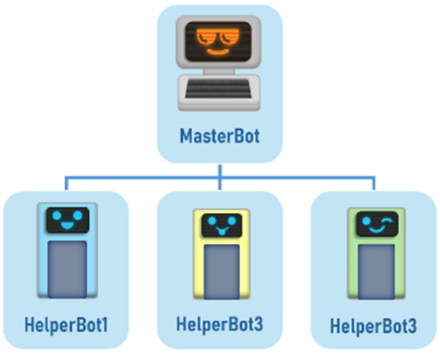

Fig. 8 Diagram of robot command and memory hierarchy.

Emotional Intelligence and Real-Life Applications The project enhances human-robot interactions by enabling robots to display appropriate emotional expressions based on the chat context. The conversational robot agents have significant potential for real-world applications in various industries, such as collaborative robotics in supermarkets, hospitals, warehouses, factories, office spaces, and care homes.

System Modularity Modularity and replaceability were emphasized through a JSON file configuration system, decoupling the robot's capabilities from the core system and allowing users to easily swap out or replace individual components without modifying the underlying code.

4 Experiments

4.1 Evaluation Approach

The evaluation approach aimed to comprehensively assess the system's performance, effectiveness, and the successful blending of classical and modern techniques to enable smooth human-robot collaboration on complex tasks. A general testing environment was developed, simulating a pastry factory operated by a swarm of robots led by a MasterBot. The simulation involved running the environment with multiple robots executing text instructions, using different input parameters for the language model.[1]

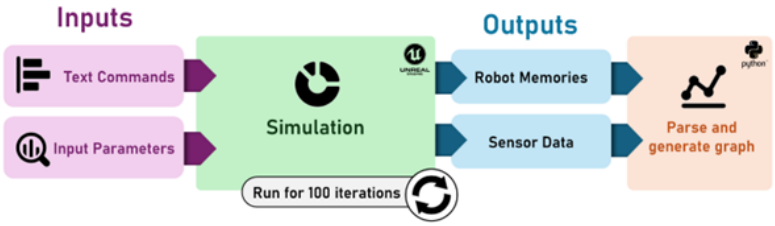

Fig. 9 Simulation Evaluation process diagram.

4.2 Evaluation Metrics

The evaluation script analyzed success rates, execution times, individual component performance, battery costs, and the language model running cost. It optimized hyperparameters through over 100 simulation runs and generated a spatial heatmap of robot locations. The analysis provided insights into the system's effectiveness, efficiency, and cost-effectiveness.

4.3 Results and Discussion

The robot agents demonstrated an encouraging overall success rate in executing natural language commands, with exceptional performance on conversational tasks. However, room for improvement exists in movement and non-stationary robot tasks. Optimal language model hyperparameters were identified, and spatial analysis revealed even robot distribution without bottlenecks. While longer text inputs slightly decreased success rates, the remarkably low cost for hundreds of API calls highlights the system's economic viability at scale.

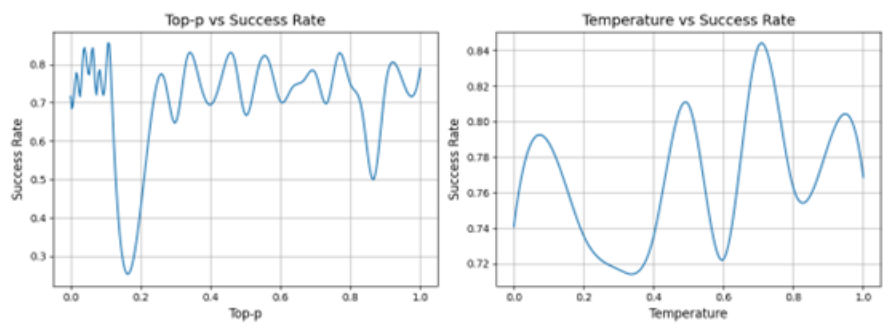

Fig. 10 Success rate plotted against isolated parameters. Top-P (left) and Temperature (right).

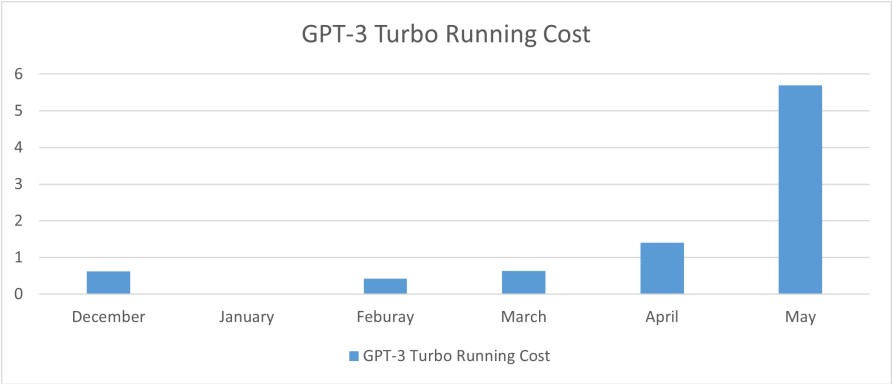

Fig. 11 Language Model API Running Cost.

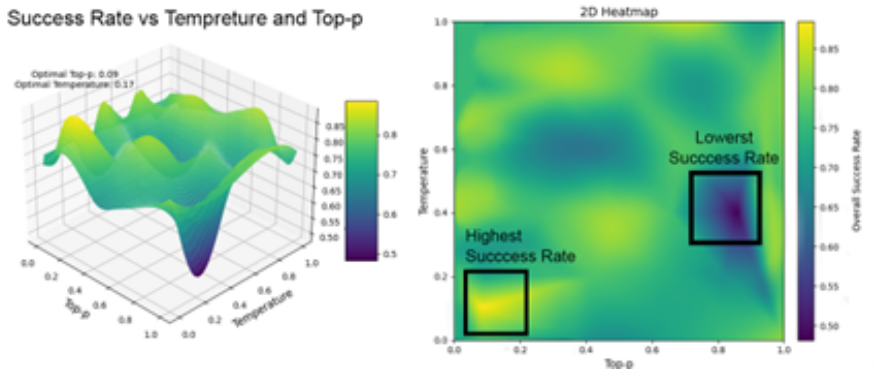

Fig. 12 Success-Rate plotted against temperature and top-p sampling values.

5 Conclusion and Future Work

The conversational robot agents project has demonstrated the promising potential of integrating large language models with robotic systems for natural language interaction and task execution. The results highlight the system's viability for real-world applications while identifying areas for improvement. As this technology matures, the integration of natural language processing and cognitive reasoning might revolutionize robot planning and integration across various domains and unlock new frontiers in automation, productivity, and human-robot interaction.

Large language models (LLMs) used in robotics can generate nonsensical or inaccurate outputs, known as hallucinations. These can lead to unintended behaviors, misinterpretations, false assumptions, and confabulation. Providing more context to the model can help mitigate hallucinations, but there is a trade-off between computational cost and the risk of insufficient grounding.

Key future enhancements include transitioning to the Robot Operating System (ROS) framework for real-world robotic hardware testing and deployment, integrating perception and navigation capabilities via real-life sensors, enabling system-generated general subroutines and action loops for task automation in factories and warehouses, and incorporating advanced planning methods like Hierarchical Task Networks (HTN) for efficient execution of complex tasks. Future work should prioritize implementing these advanced planning techniques, ROS integration, and task queuing capabilities.

Acknowledgments

We would like to acknowledge the support of [funding agency/institution] for providing the resources and funding for this research project. We also express our gratitude to [collaborators/contributors] for their valuable contributions and insights throughout the project's development.

Declarations

Some journals require declarations to be submitted in a standardised format. Please check the Instructions for Authors of the journal to which you are submitting to see if you need to complete this section. If yes, your manuscript must contain the following sections under the heading 'Declarations':

- Funding
- Conflict of interest/Competing interests (check journal-specific guidelines for which heading to use)
- Ethics approval and consent to participate
- Consent for publication
- Data availability
- Materials availability
- Code availability
- Author contribution

If any of the sections are not relevant to your manuscript, please include the heading and write 'Not applicable' for that section.

Editorial Policies for:

Springer journals and proceedings: https://www.springer.com/gp/editorial-policies

Nature Portfolio journals: https://www.nature.com/nature-research/editorial-policies

Scientific Reports: https://www.nature.com/srep/journal-policies/editorial-policies

BMC journals: https://www.biomedcentral.com/getpublished/editorial-policies

Appendix A Section title of first appendix

An appendix contains supplementary information that is not an essential part of the text itself but which may be helpful in providing a more comprehensive understanding of the research problem or it is information that is too cumbersome to be included in the body of the paper.

References

[1] Benjamin, D.P., Lonsdale, D., Lyons, D.M.: A cognitive robotics approach to comprehending human language and behaviors. In: Proceedings of the ACM/IEEE International Conference on Human-robot Interaction (2007). https://dl.acm.org/citation.cfm?doid=1228716.1228742