

LLMBot: Multi-Agent Robotic Systems for Adaptive Task Execution

Harith Ibrahim and Harith Alsafi

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

Contributing authors: harithsami01@gmail.com;
harith.alsafi@gmail.com;

Abstract

This paper presents an innovative approach integrating large language models (LLMs) with robotic systems for natural language interaction and task execution. The proposed system leverages GPT-3.5 Turbo for natural language understanding, reasoning, and high-level task planning, integrated with a simulated 3D environment in Unreal Engine. Behavior trees govern robust low-level execution of robotic actions. The system utilizes prompt engineering, multimodal input, and parameter optimization to enhance the LLM's grounding and performance. Comprehensive evaluation assesses success rates, spatial distributions, and cost-effectiveness across various scenarios. Results demonstrate promising capabilities in executing natural language commands while identifying key limitations. This research advances multi-agent collaboration and human-robot interaction by enabling more intuitive communication between humans and robots. By combining modern language models with classical symbolic AI planners, this project paves the way for more intelligent, 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

Intelligent robotic systems have the potential to revolutionize various industries, but developing sophisticated robot planning algorithms has been hindered by high costs and complexity. Recent advancements in natural language processing and large language models (LLMs) present new opportunities for more accessible and cost-effective robot planning.



Fig. 1 Diagram Showing Simplified Project Plan

Aim and Objectives The primary aim of this project is to develop an intelligent virtual robot agent capable of comprehending and executing free-form natural language instructions within a simulated 3D environment. The agent should interpret human commands, reason about the given task, and coherently carry out the necessary actions to accomplish the objective, while exhibiting adaptability to handle novel situations without extensive retraining or reprogramming.

The key objectives include:

- Develop an interactive simulation environment for virtual robot agents.
- Create virtual agents capable of executing diverse tasks based on natural language input.
- Implement an LLM-based robot behavior system for translating natural language instructions into executable behaviors.
- Optimize LLM parameters for stable robot embodiment.
- Enable robot collaboration and communication within the simulated environment.
- Evaluate system performance, running costs, and cost-effectiveness.

2 Background and related work

2.1 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. LLMs can be fine-tuned for specific tasks using techniques like prompt engineering and few-shot learning.

2.2 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.”

2.3 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. The proposed approach 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.

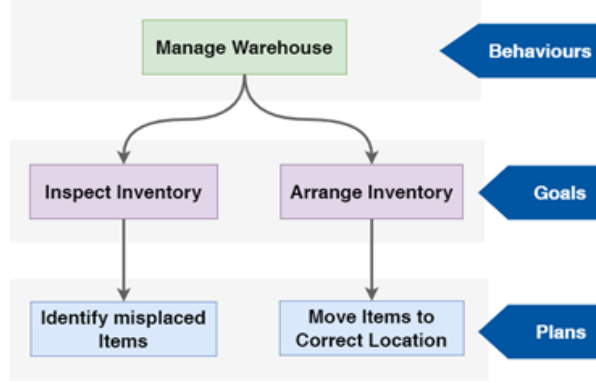


Fig. 2 Example of Behaviour Tree for planning within a warehouse scenario.

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. 3 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. 4 Screenshot showing Robot Chat User Interface.



Fig. 5 Screenshot of the simulation. Intelligent robots managing inventory in a Warehouse.



Fig. 6 Warehouse Simulation Environment in Unreal Engine 5.

Algorithm 1 Calculate $y = x^n$

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

```
1:  $y \leftarrow 1$ 
2: if  $n < 0$  then
3:    $X \leftarrow 1/x$ 
4:    $N \leftarrow -n$ 
5: else
6:    $X \leftarrow x$ 
7:    $N \leftarrow n$ 
8: end if
9: while  $N \neq 0$  do
10:  if  $N$  is even then
11:     $X \leftarrow X \times X$ 
12:     $N \leftarrow N/2$ 
13:  else [ $N$  is odd]
14:     $y \leftarrow y \times X$ 
15:     $N \leftarrow N - 1$ 
16:  end if
17: 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.

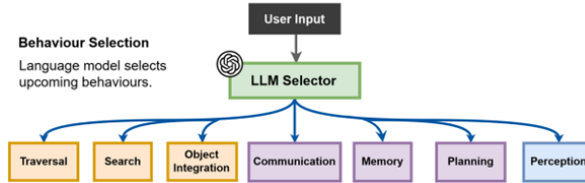


Fig. 7 Proposed approach of utilizing Language Models in Behaviour Trees.

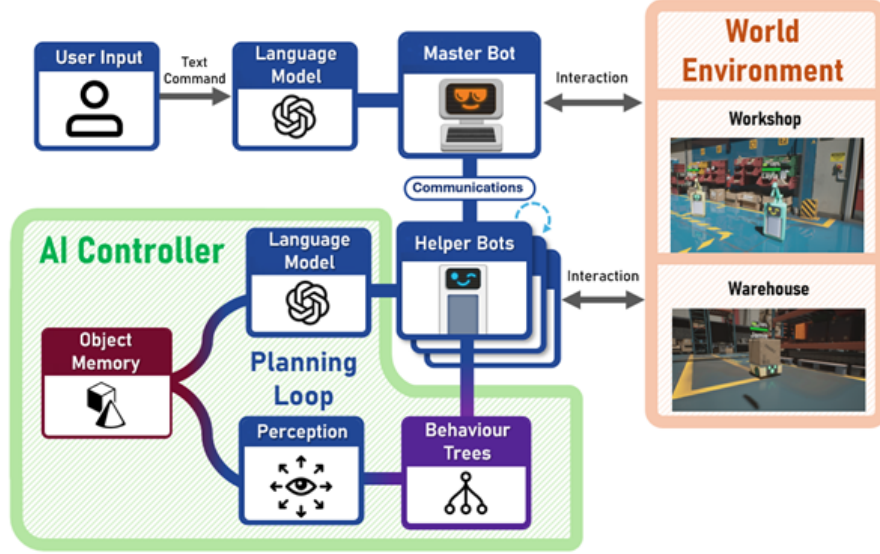


Fig. 8 Proposed Robot Cognitive Architecture.

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.

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.

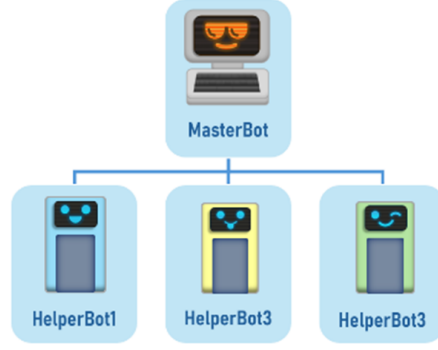


Fig. 9 Diagram of robot command and memory hierarchy.

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]

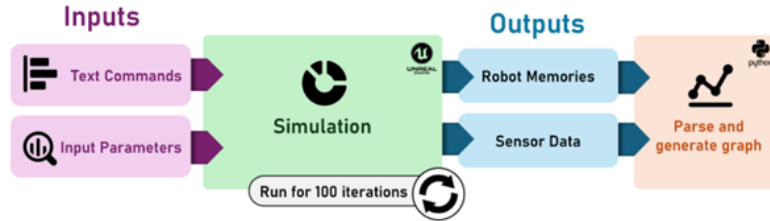


Fig. 10 Simulation Evaluation process diagram.

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.

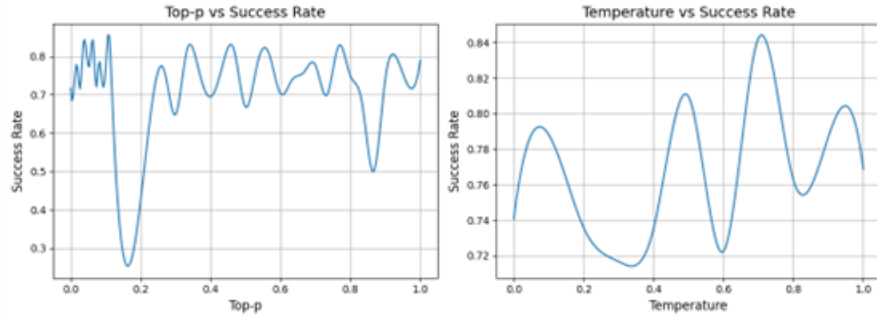


Fig. 11 Success rate plotted against isolated parameters. Top-P (left) and Temperature (right).

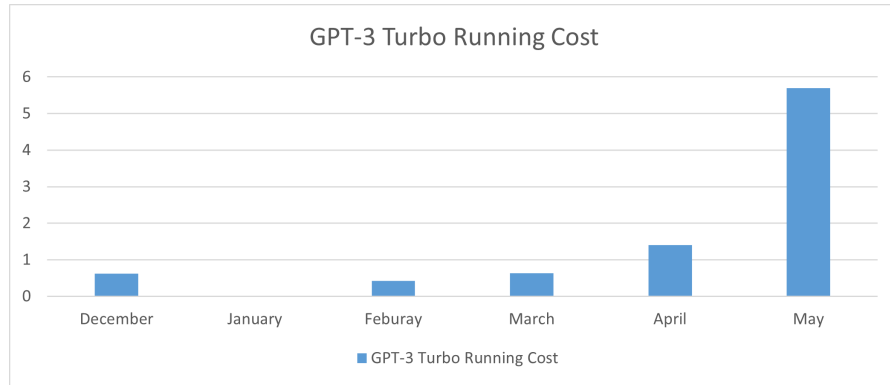


Fig. 12 Language Model API Running Cost.

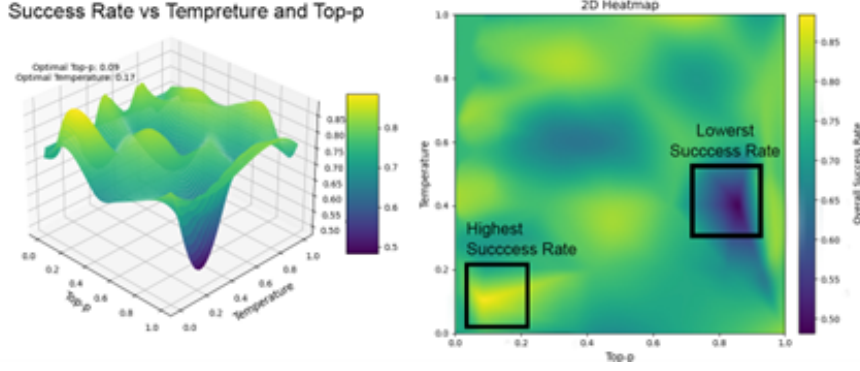


Fig. 13 Success-Rate plotted against temperature and top-p sampling values.

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>