

Long-Term Agentic Memory with LangGraph for Autonomous Task Automation and Intelligent Email Management

Udaychandra Nayak, Ben Furtado, Joel Jose, Rostan Lobo, Rhugved Mane
Department of Information Technology The Bombay Salesian Society's
Don Bosco Institute of Technology
(An Autonomous Institute affiliated to University of Mumbai)

Abstract - This paper presents a Long-Term Memory Agentic Artificial Intelligence system developed using LangGraph and LangChain frameworks. The system is capable of autonomous task execution including email classification, system automation, web search, and real-time information retrieval. The system uses a state-machine-based workflow architecture to enable intelligent decision making and multi-step tool execution. The proposed system integrates Large Language Models with automation tools to create a fully functional AI agent capable of performing complex operations efficiently while maintaining contextual awareness through long-term memory capabilities.

Keywords - Agentic AI, LangGraph, Automation, AI Agents, Email Classification, Intelligent Systems

1 INTRODUCTION

Artificial Intelligence (AI) has significantly transformed modern computing by enabling systems to perform intelligent decision making and automation. Traditional software systems rely on predefined logic and require manual intervention for performing tasks. Such systems are limited in flexibility and cannot easily adapt to dynamic user requirements. With the development of Large Language Models (LLMs), intelligent systems are now capable of understanding natural language instructions and performing complex operations with minimal human intervention.

Recent advancements in LLM-based technologies have led to the development of Agentic Artificial Intelligence systems. Unlike conventional chatbots that only generate textual responses, agentic systems are capable of interacting with external tools and executing real-world tasks. These systems can analyze user instructions, determine the required actions, and perform multi-step operations automatically. This approach enables the development of intelligent assistants that can function as practical automation platforms.

Modern computing environments require automation for tasks such as email management, system monitoring, information retrieval, and file operations. Manual execution of these tasks can be time-consuming and inefficient. Email management in particular has become increasingly

challenging due to the large volume of messages received daily. Identifying important or urgent emails requires continuous manual effort, which can be reduced through intelligent automation. Similarly, obtaining real-time information through manual web browsing can be inefficient when quick responses are required.

To address these challenges, this paper presents a ****Long-Term Memory Agentic AI system using LangGraph****, developed using LangGraph and LangChain frameworks. The proposed system integrates Large Language Models with multiple automation tools to enable autonomous task execution. The LangGraph framework provides a state-machine-based workflow architecture that enables structured decision making and reliable tool execution.

The system supports several functionalities including intelligent email classification, real-time web search, system monitoring, and automated task execution. The agent dynamically selects appropriate tools based on user instructions and performs operations autonomously. The integration of a long-term memory mechanism allows the system to maintain contextual awareness, retain historical interaction knowledge, and improve interaction continuity over extended sessions.

The objective of this work is to design and implement a practical Agentic AI platform that demonstrates the capabilities of modern AI frameworks in real-world automation scenarios. The proposed system provides a unified architecture that combines intelligent decision making with practical automation tools while leveraging long-term memory to enhance contextual reasoning and task continuity.

2 BACKGROUND

Automation has long been an important area in computer science, with traditional systems relying on predefined rules and fixed workflows. These systems are effective for repetitive tasks but lack flexibility when new requirements arise. Any modification in functionality typically requires manual reconfiguration, making such systems inefficient in dynamic environments. As a result, traditional automation tools are limited in their ability to handle complex and

evolving user requirements.

Machine learning techniques introduced improvements by allowing systems to learn patterns from data. Applications such as email filtering and spam detection use trained models to classify messages into different categories. While these approaches improve accuracy, most machine learning systems are designed for single-purpose tasks and cannot perform multiple operations within a unified framework. They also require structured data and retraining when requirements change.

Recent advancements in Large Language Models (LLMs) have significantly expanded the capabilities of intelligent systems. LLMs can interpret natural language instructions and generate meaningful responses without requiring structured input. This has enabled the development of conversational AI systems and intelligent assistants. However, most LLM-based systems function primarily as chatbots that generate responses without performing real-world actions.

Agentic Artificial Intelligence extends the capabilities of conversational AI by integrating external tools with Large Language Models. Agentic systems can analyze user instructions, determine appropriate actions, and execute tasks autonomously. This allows intelligent agents to perform operations such as retrieving information, managing files, and controlling system resources.

Frameworks such as LangChain and LangGraph have made it possible to develop structured agentic systems. LangChain provides standardized interfaces for connecting tools with Large Language Models, while LangGraph enables workflow-based execution using a state-machine architecture. These frameworks allow intelligent agents to perform multi-step reasoning, maintain contextual memory, and execute tool-based operations in a reliable and organized manner.

The proposed system builds upon these concepts by developing a multi-functional agentic platform that integrates long-term memory, email intelligence, automation tools, and real-time information retrieval within a unified architecture. The system demonstrates a practical implementation of agentic AI for real-world automation tasks.

2.1 Comparison with Existing Systems

Traditional automation systems rely on predefined rule-based workflows and require manual configuration when task requirements change. Basic conversational AI systems based on large language models primarily focus on generating textual responses and lack the ability to perform real-world operations.

The proposed LangGraph-based agentic system differs from these approaches by integrating reasoning capabilities with the execution of external tools. The system dynamically determines the sequence of operations required to complete a task and invokes appropriate tools within a structured workflow.

Feature	Traditional Systems	Proposed Agent System
Task Execution	Fixed rule-based scripts	Dynamic tool-based execution
Reasoning Capability	Limited logical flows	Multi-step reasoning using LangGraph workflows
Adaptability	Manual reconfiguration required	Automatically adapts to user instructions
Email Handling	Basic filtering	Intelligent classification and automation
Information Retrieval	Static sources	Real-time web information retrieval
Automation Scope	Single-purpose systems	Multi-functional agentic automation platform

Table 1: Comparison between traditional automation systems and the proposed LangGraph-based agentic architecture

3 DATA AND ANALYSIS

The proposed system processes multiple forms of data including user commands, email content, system information, and real-time web data. User inputs are provided in natural language and are interpreted by the Large Language Model to determine the required actions. The interpreted commands are then converted into structured tool executions through the LangGraph workflow.

Email data represents one of the primary sources of information in the system. Emails retrieved using the Gmail API are analyzed and categorized into classes such as spam, important, and urgent. The classification process is based on the content of the email including subject and message body. This allows the system to automatically organize incoming emails and assist users in identifying high-priority messages.

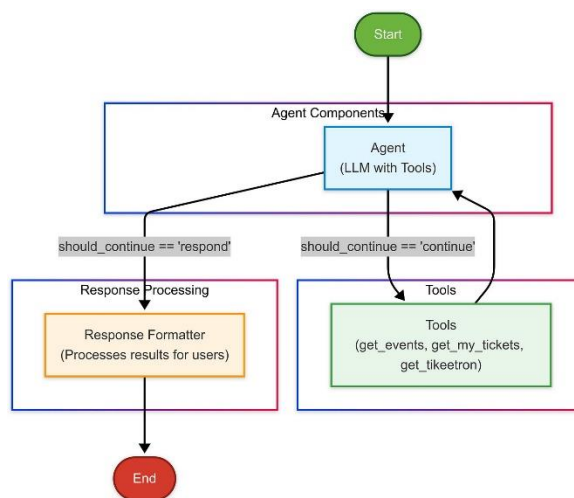


Figure 1: LangGraph workflow illustrating user input

processing, reasoning through the language model, tool selection, and execution of tasks within the agent framework.

The system also processes real-time information obtained from web sources. Web search functionality allows the agent to retrieve up-to-date information such as news, sports updates, and general knowledge queries. Search results are analyzed and relevant information is extracted from websites using automated browsing tools. This enables the system to provide accurate and timely responses.

System-level information is another important component of the data processed by the agent. The system collects information such as running processes, directory structures, and hardware statistics. This data enables the agent to perform monitoring and management tasks based on user instructions. The agent supports multiple tool-based operations including file management, system control, email handling, and automation tasks. The Large Language Model analyzes user requests and determines the appropriate sequence of tool executions required to complete each task.

Performance observations indicate that the system is capable of executing tool-based operations efficiently. Real-time queries and automation tasks are completed within short response times, and the LangGraph workflow ensures stable and organized task execution.

4 METHODOLOGY

The proposed system is designed as a long-term memory agentic Artificial Intelligence platform that integrates Large Language Models with automation tools. The architecture is based on a modular design where the intelligent agent interprets user instructions and executes tasks using appropriate tools. The overall workflow follows a structured approach consisting of input interpretation, decision making, tool selection, and execution. The architecture of the proposed system is shown below.

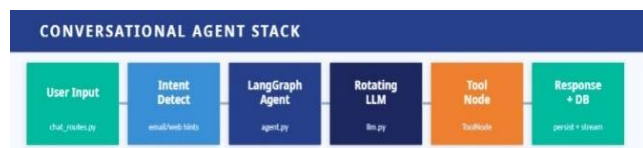


Figure 2: Conversational agent stack architecture showing user input processing, LangGraph orchestration, LLM reasoning, tool execution, and response generation.

The core intelligence of the system is implemented using the LangGraph framework, which provides a state-machine-based workflow architecture. LangGraph manages the interaction between the Large Language Model and the available tools, enabling reliable multi-step task execution. The framework allows the agent to maintain structured workflows and ensures that complex operations are executed in a controlled sequence.

LangChain is used to standardize tool definitions and manage interactions with the Large Language Model. The Large Language Model used in the system is Groq Llama-3,

which provides fast response times and strong reasoning capabilities. The model analyzes user input and determines the appropriate actions required to complete the requested task.

The system integrates multiple automation modules that can be dynamically invoked by the agent. The email automation module uses the Gmail API to fetch, read, search, and respond to emails. The email classification functionality categorizes emails into spam, important, and urgent categories based on their content.

The web information module enables real-time information retrieval. DuckDuckGo Search is used for obtaining search results, while Selenium-based automated browsing allows the system to extract information from dynamic web pages. This enables the agent to provide up-to-date responses to user queries. System automation features allow the agent to perform operations such as monitoring processes, accessing directories, and executing commands. File management operations such as reading, writing, moving, and deleting files are also supported. These functionalities allow the agent to perform practical automation tasks within the operating system environment.

The backend of the system is implemented using the Flask framework, which provides API endpoints and system control functionality. Flask-SocketIO enables real-time communication between the user interface and the server. Secure authentication is implemented using JWT tokens and Bcrypt hashing.

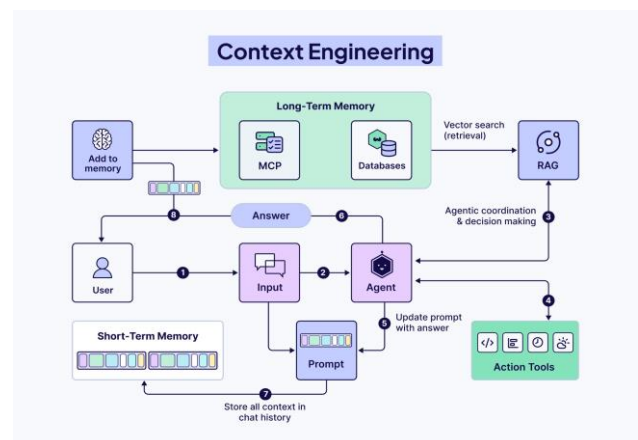


Figure 3: Long-term agentic memory architecture showing interaction between the language model, contextual memory storage, and knowledge retrieval components.

SQLite is used as the primary database for storing user data, chat history, and system logs. The system is deployed on a Linux-based server environment with support for automation tools and headless browser execution.

The overall workflow of the system can be summarized as follows. The user provides a command in natural language, which is interpreted by the Large Language Model. LangGraph determines the sequence of actions required and selects the appropriate tools. The tools execute the requested operations and the results are returned to the user interface.

5 RESULTS AND DISCUSSION

The proposed long-term memory agentic AI system using LangGraph was successfully implemented and evaluated for various automation tasks including email classification, real-time information retrieval, and system-level operations. The intelligent agent demonstrated the ability to interpret natural language commands and execute appropriate tools through the LangGraph-based workflow architecture. The system maintained stable operation while performing multiple tasks and switching between different tool modules.

The email intelligence module was tested using different categories of emails including promotional messages, normal communications, and high-priority messages. The agent successfully classified emails into spam, important, and urgent categories based on the content of the subject and message body. This functionality allows users to quickly identify relevant messages and reduces the effort required for manual email sorting. Email fetching, searching, and reply operations were also performed successfully using the integrated Gmail API.

The real-time information retrieval module was tested using various search queries including sports updates, general information, and current events. The system was able to retrieve relevant information from web sources using the integrated search and browsing tools. The agent provided responses within a short time, demonstrating the effectiveness of real-time web integration.

System automation features were evaluated by testing operations such as process monitoring, directory browsing, and command execution. File management tasks including reading and writing files were also successfully executed. The agent was able to select appropriate tools and perform operations reliably within the system environment.

Overall, the results demonstrate that the proposed system is capable of performing intelligent automation tasks efficiently. The LangGraph workflow architecture ensured structured execution and reliable coordination between the Large Language Model and the integrated tools. The inclusion of long-term memory mechanisms also enabled improved contextual understanding and continuity across interactions.

6 CONCLUSION

This paper presented a Long-Term Memory Agentic Artificial Intelligence system developed using the LangGraph and LangChain frameworks for autonomous task execution and intelligent automation. The proposed system integrates Large Language Models with multiple automation tools to create an intelligent agent capable of performing practical operations across different domains.

The system successfully demonstrated several key functionalities including intelligent email classification, real-time web information retrieval, system monitoring, and automated task execution. Through the use of the LangGraph state-machine workflow architecture, the agent was able to perform structured reasoning and coordinate multiple tools in a controlled and reliable manner. This

workflow-based approach enabled the system to handle complex multi-step tasks efficiently while maintaining stability during execution.

A significant contribution of the proposed system is the integration of long-term memory capabilities within the agentic architecture. The memory mechanism enables the system to retain contextual information across interactions, allowing the agent to maintain continuity and improve its understanding of user instructions over time. This capability enhances the effectiveness of the agent in real-world automation environments where maintaining contextual awareness is essential for intelligent decision making.

The modular architecture of the system also allows additional automation tools and services to be integrated easily. This flexibility makes the platform scalable and adaptable for different applications such as enterprise automation, intelligent assistants, and system management tools. By combining language model reasoning with structured workflows and tool-based execution, the proposed system demonstrates the practical potential of agentic AI technologies.

Future work may focus on improving the long-term memory mechanisms, expanding the range of automation capabilities, and enhancing system scalability. Additional features such as voice-based interaction, mobile platform integration, and more advanced reasoning strategies may further improve the usability and effectiveness of the system. These enhancements can contribute toward the development of more autonomous and intelligent agentic systems capable of supporting complex real-world workflows.

REFERENCES

- [1] T. B. Brown et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [2] J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [3] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.
- [4] A. Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models," *arXiv preprint arXiv:2210.03629*, 2022.
- [5] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th Edition, Pearson, 2020.
- [6] Y. Liu et al., "Pre-trained Language Models for Natural Language Processing: A Survey," *ACM Computing Surveys*, 2021.
- [7] OpenAI, "GPT-4 Technical Report," *arXiv preprint arXiv:2303.08774*, 2023.
- [8] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL-HLT*, 2019.