INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 7, July 2026 | ISSN: 2456-4184 | IJNRD.ORG

Design and Prototyping of a Multi-Agent AI System for Intelligent Task Automation

Bantupalli Renuka Sai[1], Boddepalli Chandini[2], Bapatla Shanmuka Sai Satwick[3], Bankuru Lohith Kumar[4], Dasari Mounika[5]

[1]Department of Computer Science & Engineering - Data Science
Anil Neerukonda Institute of Technology and Sciences
Visakhapatnam, India
renukabantupalli.csd@anits.edu.in
boddepallichandini.22.csd@anits.edu.in
bapatlashanmukasaisatwick.22.csd@anits.edu.in
bankurulohithkumar.22.csd@anits.edu.in
dasarimounika.22.csd@anits.edu.in

Abstract: Large Language Models (LLMs) have enhanced the performance of artificial intelligence systems by advancing natural language processing and generation. However, conventional single agent systems often face limitations in handling complex queries that require task decomposition, reasoning, and information retrieval. In light of such constraints, this paper presents a solution based on a multi-agent architecture for query processing intelligently with LangChain and LangGraph. The proposed system consists of multiple specialized agents including a Planner Agent, Retriever Agent, and Responder Agent that collaboratively process user queries. The Planner Agent decomposes complex tasks into smaller subtasks, the Retriever Agent gathers relevant contextual information, and the Responder Agent generates accurate responses using large language models. LangGraph is used to manage the workflow and communication between agents, ensuring efficient coordination and scalability. The system is implemented using Python and evaluated through multiple query scenarios to analyze response accuracy and efficiency. Experimental results demonstrate that the proposed multi-agent architecture improves response accuracy and reasoning capability compared to traditional single-agent systems. The framework provides a scalable solution for building advanced AI assistants capable of solving complex real-world problems.

Keywords: Multi-Agent Systems, LangChain, LangGraph, Large Language Models, AI Agents, Intelligent Query Processing

1 Introduction

The field of artificial intelligence has made great progress due to the emergence of large language models, which can comprehend and generate text like humans. Such technologies find application in various areas, including chatbots, virtual assistants, and information management systems. However, many existing AI systems rely on a single-agent architecture where one model is responsible for processing the entire user request, which can lead to limitations when handling complex queries that require reasoning and information retrieval [9]. To deal with these problems, multi-agent systems are created wherein several intelligent agents work together to carry out various tasks. The intelligent agents are assigned different tasks like task planning, information gathering, conversation tracking, and response formation. This collaborative approach improves the system's ability to handle complex queries more efficiently and accurately [3][5].
In this research, a multi-agent framework is proposed for intelligent query processing using LangChain and LangGraph. The system integrates multiple specialized agents including a Planner Agent, Retriever Agent, and Responder Agent that work together to process user queries. The proposed approach aims to improve response accuracy, reasoning capability, and overall system efficiency compared to traditional single-agent architectures [10].

2 Related Work

Recent research has focused on developing intelligent systems using multi-agent architectures and large language models. These systems enable multiple agents to collaborate in solving complex tasks and improving reasoning capabilities.
Previous studies have proposed multimodal agent frameworks capable of integrating different data sources such as text, images, and external knowledge bases to perform complex tasks efficiently [1]. Other research has presented taxonomies for autonomous LLM-powered multi-agent architectures, which categorize different design strategies for agent collaboration and decision-making [2].
Many approaches have also been investigated for collaborative agent communication and hierarchical coordination strategies to enhance task automation and reasoning abilities within artificial intelligence [4][6]. Furthermore, surveys of LLM-based multi-agent systems recently conducted emphasize the increasing significance of agent collaboration, memory, and workflow structuring when designing AI assistants. [10].

3 Methodology

As suggested in the system, the multi-agent methodology is used in order to efficiently process the queries. Using the multi-agent strategy, the tasks are assigned to different agents that can efficiently execute a particular role in the processing of the query. [3][5].
The framework consists of four primary agents: Planner Agent, Retriever Agent, Memory Agent, and Responder Agent. The Planner Agent analyzes the user query and determines the sequence of actions required for processing. The Retriever Agent gathers relevant information from available knowledge sources, while the Memory Agent maintains contextual information and conversation history to support coherent responses [6][7].
Finally, the Responder Agent generates the final response by integrating the information obtained from other agents. The coordination between agents is implemented using modern AI frameworks such as LangChain and LangGraph, which enable structured communication and workflow management among agents [4].

System Architecture: Architecture of the Proposed Multi-agent System The proposed architecture for the multi-agent system is aimed at ensuring that the user queries are processed in an efficient manner via interaction amongst several intelligent agents. Under the proposed architecture, the whole process is broken down into several sub-tasks, with each individual agent performing a particular function within the process. [3][8].
The architecture consists of four main agents: Planner Agent, Retriever Agent, Memory Agent, and Responder Agent. When a user submits a query, the Planner Agent first analyzes the request and determines the sequence of operations required to process the query. Based on this plan, the Retriever Agent collects relevant information from available knowledge sources. The Memory Agent maintains contextual information and stores conversation history to support accurate and consistent responses [10].
After gathering the required information, the Responder Agent generates the final response by integrating the outputs obtained from other agents. The interaction and coordination between these agents are managed using LangChain and LangGraph, which enable structured communication and efficient workflow management within the multi-agent system [6].

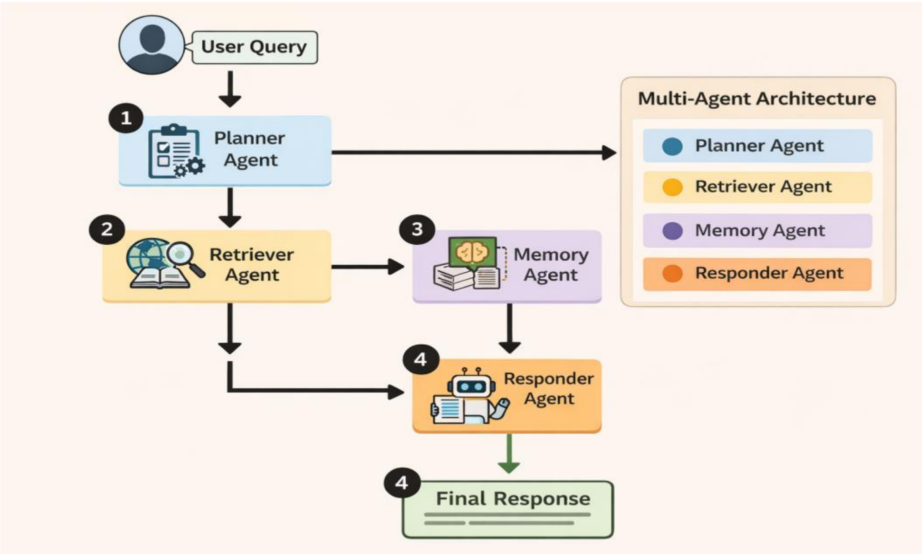

Fig. 1. Proposed Multi-Agent Architecture for Intelligent Query Processing.

4 Experimental Results

4.1 Implementation setup

The multi-agent system that was proposed, implemented using the Python programming language. The system integrates large language models with agent-based workflows using LangChain and LangGraph. These frameworks enable structured communication and coordination between the Planner Agent, Retriever Agent, Memory Agent, and Responder Agent. The system was developed and tested using the Visual Studio Code environment.

4.2 Performance Evaluation

The proposed multi-agent system was tested with several user queries to evaluate its ability to generate accurate and context-aware responses. The Planner Agent analyzes the query and coordinates with the Retriever, Memory, and Responder agents to produce the final response.The performance of the proposed system is evaluated using accuracy and response time metrics. **Fig. 2** shows the accuracy comparison with existing systems, while **Fig. 3** illustrates the response time for different query complexities.

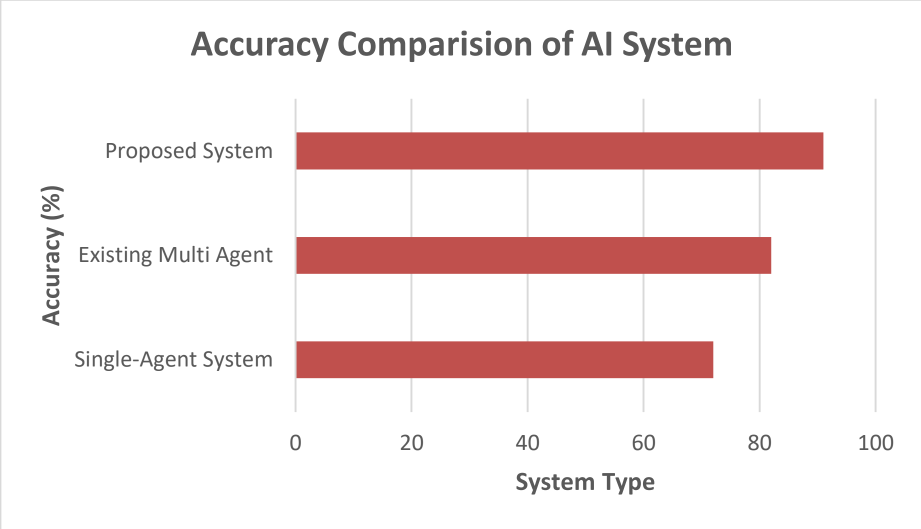

Fig. 2. Accuracy comparison between AI systems.

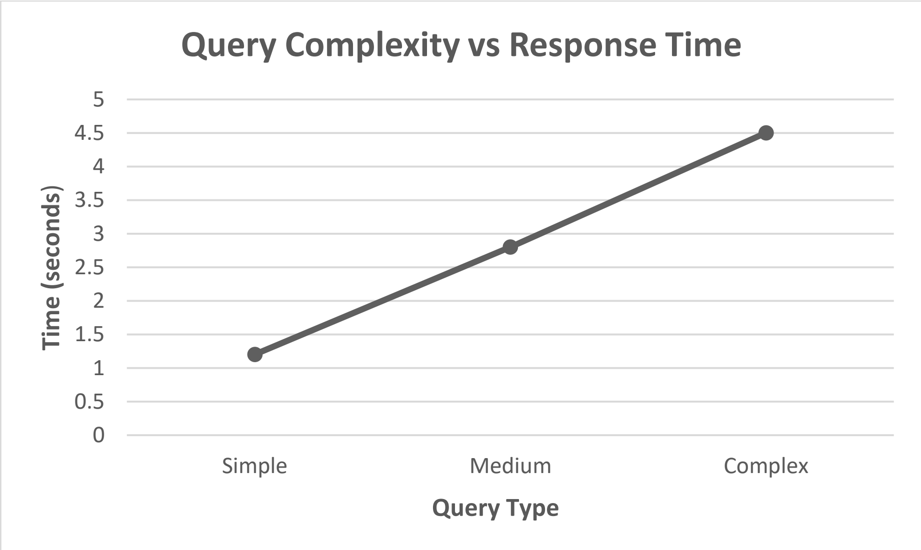

Fig. 3. Response time based on query complexity.

4.3 Accuracy Analysis

Model / Framework	Multi-Agent Support	Task Planning	Memory Handling	Information Retrieval	Response Accuracy
OmniNova Framework	Yes	Partial	Limited	Yes	Moderate
BMW Agents Framework	Yes	Yes	Limited	Yes	Moderate
LLM Multi-Agent Taxonomy	Conceptual	No	No	Limited	Moderate
Proposed System	Yes	Yes	Yes	Yes	High

Fig. 4. Accuracy Table.

Fig. 4 presents the accuracy comparison of different systems based on evaluation criteria such as task planning, memory handling, and information retrieval. The proposed system achieves higher accuracy due to efficient coordination between agents and better context management.

5.Conclusion And Future Work

This paper presents a multi-agent AI framework for intelligent query processing using LangChain and LangGraph. The proposed system integrates multiple agents responsible for task planning, information retrieval, memory management, and response generation. By distributing tasks among specialized agents, the system improves coordination and response quality. Experimental results show that the multi-agent architecture provides better accuracy and efficiency compared to traditional single-agent systems [3][5]. The proposed framework demonstrates the potential of multi-agent collaboration in developing more intelligent and scalable AI applications [10].

6.References

1. OmniNova: A General Multimodal Agent Framework, arXiv preprint arXiv:2503.20028 (2025)
2. Balancing Autonomy and Alignment: A Multi-Dimensional Taxonomy for Autonomous LLM-Powered Multi-Agent Architectures, arXiv preprint arXiv:2310.03659 (2023)
3. Multi-Agent Collaboration: Harnessing the Power of Intelligent LLM Agents, arXiv preprint arXiv:2306.03314 (2023)
4. Talk Structurally, Act Hierarchically: A Collaborative Framework for LLM Multi-Agent Systems, arXiv preprint arXiv:2502.11098 (2025)
5. Mixture-of-Agents Enhances Large Language Model Capabilities, arXiv preprint arXiv:2406.04692 (2024)
6. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework, Microsoft Research, 2023
7. AutoAgents: A Framework for Automatic Agent Generation, Peking University, 2023
8. Coordinated LLM Multi-Agent Systems for Collaborative Question-Answer Generation, Knowledge-Based Systems, Elsevier (2025)
9. Language Models are Few-Shot Learners, Advances in Neural Information Processing Systems (NeurIPS), 33, 1877–1901 (2020)
10. Large Language Model based Multi-Agents: A Survey of Progress and Challenges, arXiv preprint arXiv:2402.01680 (2024)