Research Article

Exploration of LLM Multi-Agent Application Implementation Based on LangGraph+CrewAI

Zhihua Duan[1], **Jialin Wang**[1]

1. Independent researcher

With the rapid development of large model technology, the application of agent technology in various fields is becoming increasingly widespread, profoundly changing people's work and lifestyles. In complex and dynamic systems, multi-agents achieve complex tasks that are difficult for a single agent to complete through division of labor and collaboration among agents. This paper discusses the integrated application of LangGraph and CrewAI. LangGraph improves the efficiency of information transmission through graph architecture, while CrewAI enhances team collaboration capabilities and system performance through intelligent task allocation and resource management. The main research contents of this paper are: (1) designing the architecture of agents based on LangGraph for precise control; (2) enhancing the capabilities of agents based on CrewAI to complete a variety of tasks. This study aims to delve into the application of LangGraph and CrewAI in multi-agent systems, providing new perspectives for the future development of agent technology, and promoting technological progress and application innovation in the field of large model intelligent agents.

Corresponding authors: Zhihua Duan, duanzh.sh@chinatelecom.cn; Jialin Wang, jialinwangspace@gmail.com

I. Introduction

Driven by the rapid advancements in artificial intelligence technology, the application of agents in various fields is increasingly growing, profoundly affecting everyone's work and lifestyle. The application scope of agent technology is broad; it can autonomously perceive the environment, conduct data analysis, and make decisions, thereby significantly enhancing efficiency and optimizing resource allocation. In complex and dynamic systems, the introduction of multi-agent systems enables multiple agents to collaborate and complete complex tasks that are difficult for a single agent to achieve.

The key advantage of multi-agent systems lies in their task decomposition capabilities, achieving goals through the collaborative action of agents, which not only enhances the system's flexibility and adaptability but also improves its generalization ability. In environments characterized by uncertainty and dynamic changes, the ability of agents to collaborate and divide tasks is particularly crucial.

 doi.org/10.32388/R27SW4

This paper primarily investigates two issues: (1) designing the architecture of agents based on LangGraph for more precise control; (2) enhancing the capabilities of agents based on CrewAI to accomplish different tasks. The aim is to delve into the application of AI multi-agent systems, explore the technological advantages brought about by the combination of LangGraph and CrewAI, and their potential application value in various fields. Through the analysis and research of these technologies, it is expected to provide new perspectives and ideas for the future development of agent technology, thereby promoting technological progress and application innovation in the field of large model intelligent agents.

II. Related Work

Autonomous agents are typically responsible for specific roles to accomplish various tasks. MetaGPT[1], ChatDev[2], and self-collaboration[3] pre-set various roles and corresponding responsibilities to foster collaboration among agents.

Autonomous agents based on Large Language Models (LLMs) incorporate mechanisms from human memory processes. RLP[4] is a conversational agent that maintains an internal state for both parties in a dialogue, achieving the agent's short-term memory. SayPlan[5] is an agent designed for task planning and design. When faced with complex tasks, break them down into simpler subtasks and solve them. The Chain of Thought (CoT) [6] inputs reasoning steps for solving complex problems into the prompt.[7] proposes an intelligent personalized digital banking assistant based on LangGraph and Chain of Thoughts (COT), leveraging Large Language Models (LLM) and a multi-agent framework to enhance task efficiency. [8] improves advanced question-answering systems based on Retrieval-Augmented Generation (RAG) by leveraging graph technology, to overcome the limitations of existing RAG models and develop high-quality artificial intelligence services.

In this study, autonomous agents based on LLM can leverage the framework capabilities of LangGraph and CrewAI to automatically perform various tasks, endowing Agents with the ability to complete specific tasks, forming a comprehensive application system framework.

III. Methodology

A. Introduction to LangGraph

LangGraph is a framework designed for constructing multi-agent applications, enabling developers to utilize large language models (LLMs) to create agents and multi-agent workflows. Compared to other LLM frameworks, LangGraph offers the benefits of loops, controllability, and persistent memory. LangGraph provides granular control over the application's processes and states, enabling the creation of reliable agents that support advanced human-computer interaction and memory capabilities. The flexible framework of LangGraph supports various control flows—single-agent, multi-agent, hierarchical, sequential and can robustly manage complex real-world scenarios.

B. CrewAI Framework

CrewAI is an open-source framework designed to coordinate AI agents with role-playing and autonomous operations to facilitate cooperation among agents in solving complex problems. It allows developers to define AI agents with specific roles, objectives, and tools. The main building blocks of CrewAI include Agent, Task, Tool, and Crew, offering a rich set of

features that can be freely selected and combined according to specific needs to create multi-agent systems. CrewAI supports various APIs such as OpenAI and Ollama, and it has key features like role-customized agents, automatic task delegation, and flexibility in task management. CrewAI is aimed at handling complex tasks, such as multi-step workflows, decision-making, and problem-solving.

An Agent intelligent entity is an autonomous unit capable of performing tasks, making decisions, and communicating with other agents. In the CrewAI framework, a Task refers to the specific work carried out by an Agent, including details such as description, executing agents, and required tools. It supports multi-agent collaboration and optimizes team cooperation and efficiency through the process orchestration of the Crew.

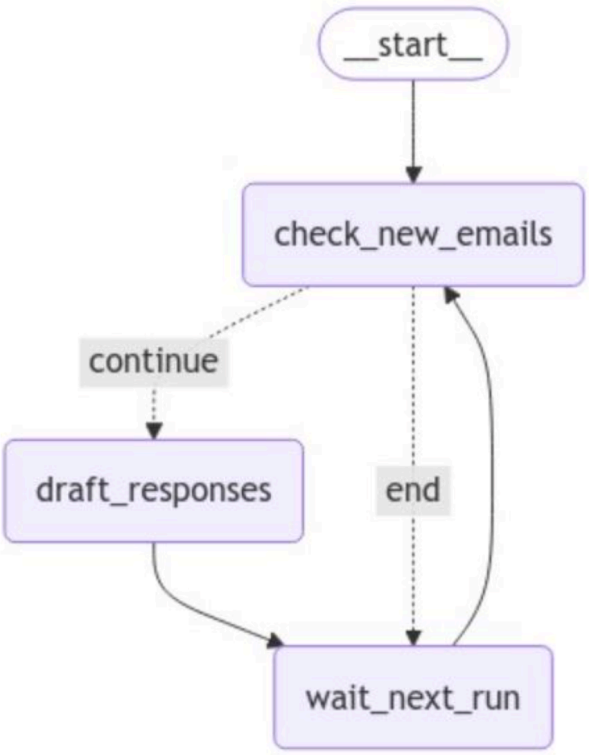

Figure 1. Illustration of the email case based on LangGraph +CrewAI.

C. Integrating LangGraph with CrewAI

The integration of LangGraph+CrewAI framework provides powerful tools for complex task management, optimizing task execution and multi-agent systems through flexible workflows, inter-agent collaboration, and graph-structured management. It supports customized development by integrating with existing tools, meeting the evolving needs of AI applications. CrewAI ensures process efficiency through clear task allocation and role definition. Additionally, seamless integration with LangChain allows developers familiar with the framework to easily integrate independent agents, further enhancing the framework's appeal.

Taking the example of the automatic email composition and sending case provided on the CrewAI official website, the LangGraph+CrewAI framework breaks down complex tasks into manageable steps and automates their execution.

As shown in Figure 1, with LangGraph, it is possible to clearly define and visualize the workflow for writing and sending emails, including checking new emails, composing new emails, and waiting for the next run.

As shown in Figures 2 to 4 ,this is a code example based on CrewAI+LangGraph. By integrating CrewAI with LangGraph, it becomes easy to implement functionalities such as email checking, composing, and automatic sending based on large models.

```python
from dotenv import load_dotenv
load_dotenv()
from langgraph.graph import StateGraph
from .state import EmailsState
from .nodes import Nodes
from .crew.crew import EmailFilterCrew

class WorkFlow():
    def __init__(self):
        nodes = Nodes()
        workflow = StateGraph(EmailsState)
        workflow.add_node("check_new_emails", nodes.check_email)
        workflow.add_node("wait_next_run", nodes.wait_next_run)
        workflow.add_node("draft_responses", EmailFilterCrew().kickoff)
        workflow.set_entry_point("check_new_emails")
        workflow.add_conditional_edges(
                "check_new_emails",
                nodes.new_emails,
                {
                    "continue": 'draft_responses',
                    "end": 'wait_next_run'
                }
        )
        workflow.add_edge('draft_responses', 'wait_next_run')
        workflow.add_edge('wait_next_run', 'check_new_emails')
        self.app = workflow.compile()
```

Figure 2. Code example based on LangGraph + CrewAI.

```python
class EmailFilterAgents():
    def __init__(self):
        self.gmail = GmailToolkit()

    def email_filter_agent(self):
        return Agent(
            role='Senior Email Analyst',
            goal='Filter out non-essential emails like newsletters and promotional content',
            backstory=dedent("""\
                As a Senior Email Analyst, you have extensive experience in email content analysis.
                You are adept at distinguishing important emails from spam, newsletters, and other
                irrelevant content. Your expertise lies in identifying key patterns and markers that
                signify the importance of an email."""),
            verbose=True,
            allow_delegation=False
        )
```

Figure 3. Agent example based on CrewAI.

```python
class EmailFilterTasks:
    def filter_emails_task(self, agent, emails):
        return Task(
            description=dedent(f"""\
                Analyze a batch of emails and filter out
                non-essential ones such as newsletters, promotional content and notifications.

                Use your expertise in email content analysis to distinguish
                important emails from the rest, pay attention to the sender and avoind invalid emails.

                Make sure to filter for the messages actually directed at the user and avoid notifications

                EMAILS
                -------
                {emails}

                Your final answer MUST be a the relevant thread_ids and the sender, use bullet points.
                """),
            agent=agent
        )
```

Figure 4. Task example based on CrewAI.

IV. Application Case Practice

A. Development Environment

In this study, we utilized the following development environment components to apply the LangGraph+CrewAI framework:

- Multi-agent: CrewAI, for organizing and coordinating collaborative work among AI agents.
- Graph workflow: LangGraph, used for managing state sharing and process control of graph nodes.
- Workflow tracking: LangSmith, for monitoring and auditing the execution of workflows.

- Embedding technology: ollama embedding model, utilizing embedded vector models to retrieve work order text.
- Vector database: Fasiss is used for storing and retrieving vector data, accelerating data access speed.

B. Application Cases

As shown in Figure 5, based on the LangGraph+CrewAI framework, we have explored and implemented a multi-agent collaborative application that integrates code generation and code review capabilities. By achieving real-time sharing of status data and feedback mechanisms, the efficiency of code generation has been improved.

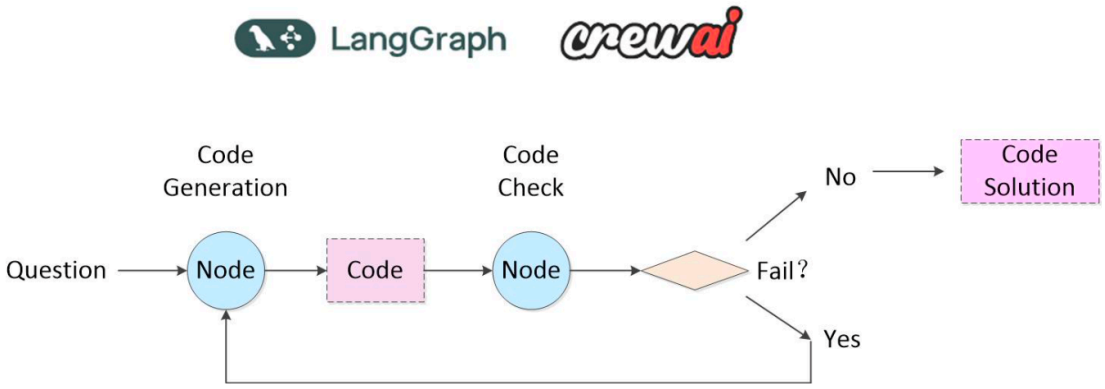

Figure 5. A code generation case based on LangGraph+CrewAI.

As shown in Figure 6, based on the CrewAI+LangGraph framework, we have explored and implemented a case for ticket auditing and forwarding features. By conducting an in-depth analysis of ticket text information, we have built multiple intelligent agents that can more accurately understand the content of the tickets, thereby enhancing the efficiency of ticket processing.

doi.org/10.32388/R27SW4

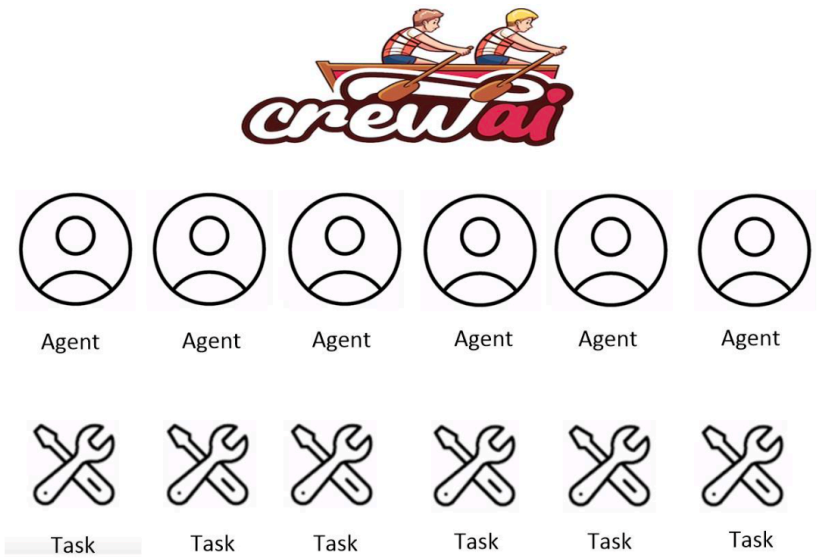

Figure 6. Case study of automatic processing of work orders based on LangGraph+CrewAI

V. Conclusion

This article explores the combined application of LangGraph and CrewAI frameworks, demonstrating the powerful capabilities of these two frameworks in building complex multi-agent systems and workflow management. Through case analysis, the LangGraph+CrewAI framework has advantages in task management, inter-agent collaboration, graph workflow implementation, and integration with existing tools. This integrated approach not only improves the efficiency of task execution but also enhances the system's flexibility and scalability through real-time status data sharing and feedback mechanisms. Our research results indicate that the combination of LangGraph and CrewAI provides a powerful toolkit for developing advanced AI applications, especially in scenarios that require handling complex tasks and multi-agent collaboration. By integrating with LangChain, existing independent agents can be easily incorporated into the CrewAI framework, providing developers with a unified platform for building and managing complex AI workflows.

References

1. ^*Zhuge M, Hong S, et al. Metagpt: Meta programming for multi-agent collaborative framework. International Conference on Learning Representations. 2024.*
2. ^*Wei L, Chen Q, et al. "Chatdev: Communicative agents for software development." In Proceedings of the 62nd Annual Meetin g of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15174–15186.*
3. ^*Xue J, Yihong D, et al. Self-collaboration code generation via chatgpt. CoRR. 2023; abs/2304.07590.*
4. ^*Fischer KA (2023). "Reflective linguistic programming (rlp): A stepping stone in socially-aware agi (socialagi)". CoRR. abs/2 305.12647.*

5. ^*Haviland J, Rana K, et al. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. CoRR. 2023.*
6. ^*Wang X, Wei J, et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Informati on Processing Systems. **35**:24824–24837, 2022.*
7. ^*Saha S, Arafat ME, et al. An intelligent LLM-powered personalized assistant for digital banking using LangGraph and chain of thoughts. IEEE 22nd Jubilee International Symposium on Intelligent Systems Informatics. 2024:625–630.*
8. ^*Jeong C. "A study on the implementation method of an agent-based advanced rag system using graph". CoRR. **abs/2407.199 94**, 2024.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.