From Swarm to Society: A Multi-Agent Adaptive Framework for Robust LLM Collaboration Beyond Standalone LLMs

Giovanpaolo Vrenna
Giovanpaolo.vrenna@keio.jp

Shigeki Noguchi
noguchi_shigeki@keio.jp

Naoki Watson
naokiwatson@keio.jp

Keita Fujie
fk77663@keio.jp

Abstract

This report presents a novel approach to enhancing the accuracy and contextual understanding of large language models (LLMs) through the integration of swarm intelligence principles. We have developed a custom multi-agent framework modeled after planetary systems, where autonomous agents interact dynamically to generate and refine outputs collaboratively. Initially, our implementation was based on OpenAI's Swarm framework, which allowed us to establish a solid foundation and introduce significant features into the codebase. However, recognizing the need for greater flexibility and customization, we have begun migrating to Microsoft's Autogen framework.

The transition to Autogen enables advanced functionalities such as dynamic role allocation, nested group communication, and real-time decision-making, which were limitations in the Swarm framework. Our current focus is on transferring the robust features developed in the Swarm codebase to the Autogen environment. This migration aims to enhance iterative collaboration between agents, improve consensus-driven decision-making, and ultimately boost problem-solving accuracy and adaptability.

By leveraging the principles of decentralized intelligence and swarm behaviors, our framework could demonstrate the potential to overcome limitations inherent in standalone LLMs. Preliminary results suggest that this multi-agent approach paves the way for more robust and adaptive AI systems capable of handling complex tasks through collective intelligence.

1 Introduction

Large language models (LLMs) like OpenAI's GPT-4 have set new benchmarks in natural language processing (NLP), excelling across diverse applications. Despite their impressive capabilities, LLMs exhibit notable limitations, including inconsistent reasoning, vulnerability to contextual misunderstandings, and challenges in generating precise, domain-specific insights. Addressing these gaps necessitates innovative approaches to enhance both the accuracy and reliability of LLM outputs.

Swarm intelligence, inspired by the collective problem-solving behaviors of biological systems such as ant colonies and bird flocks, provides a compelling paradigm to overcome these challenges. By fostering decentralized collaboration among autonomous entities, swarm intelligence achieves emergent solutions that surpass the capabilities of individual members. This concept aligns with the inherent limitations of LLMs, offering a promising framework for iterative reasoning and collective problem-solving.

In this project, we have developed a novel multi-agent system that operationalizes swarm intelligence principles to enhance the contextual understanding and problem-solving capabilities of LLMs. Each agent functions autonomously, contributing incrementally to a larger collective decision-making process. To further structure collaboration, we have organized agents into "planets," independent problem-solving units capable of parallel processing. A central hub facilitates interplanetary communication, enabling seamless integration of diverse solutions to complex queries.

The project began with the OpenAI Swarm framework, which provided a solid foundation for the development of the multi-agent architecture. Swarm's capabilities allowed us to define agent roles and manage parallel interactions effectively. However, its rigid architecture posed challenges in dynamic task allocation and inter-agent communication, prompting a transition to Microsoft's Autogen framework. Autogen introduces advanced features such as dynamic role reassignment, nested group communication, and greater scalability, aligning with our goal of creating a flexible and adaptive multi-agent system.

This report details our progress, including the conceptual underpinnings of the project, the development and refinement of our codebase, and the ongoing migration from Swarm to Autogen. By leveraging the principles of swarm intelligence, our work aims to bridge the limitations of standalone LLMs and pave the way for more robust, decentralized AI systems.

2 Background

Swarm intelligence, inspired by collective behaviors observed in natural systems like ant colonies and bird flocks, represents a decentralized approach to problem-solving. These systems exhibit emergent intelligence, where the combined actions of individuals yield solutions that surpass the capabilities of any single member. Translating these principles to artificial systems has driven advancements in fields such as robotics, optimization algorithms, and distributed AI. Our project builds on these foundations, adapting swarm intelligence concepts to enhance the performance and reliability of large language models (LLMs).

2.1 OpenAI Swarm

OpenAI's Swarm framework served as the initial platform for implementing our multi-agent system [1]. It provided a robust environment for defining agent roles, parallelizing agent interactions, and managing task workflows. This foundational framework allowed us to explore the feasibility of swarm intelligence in refining LLM outputs. However, Swarm's rigid architecture imposed significant limitations, particularly in dynamic task allocation and inter-agent communication. These constraints led us to evaluate alternative frameworks that could offer greater flexibility and adaptability.

2.2 Microsoft Autogen

Microsoft's Autogen framework emerged as a superior alternative, aligning closely with the needs of our project. Autogen introduces advanced capabilities, including dynamic role reassignment, nested group communication, and real-time interaction management [2]. Its "Society of Mind" model, inspired by Marvin Minsky's conceptual framework [3], enables intelligence to emerge from the interactions of independent agents. By adopting Autogen, we gained the flexibility to design a more scalable and adaptive system, facilitating the seamless integration of dynamic, consensus-driven decision-making processes.

3 Methodology

Our methodology integrates swarm intelligence principles with the computational strengths of large language models (LLMs) to create a robust multi-agent framework. This section outlines the core architectural and procedural elements of our approach.

3.1 Dynamic Multi-Agent Architecture

At the heart of our framework lies a dynamic multi-agent architecture designed to mimic the decentralized intelligence observed in natural systems. Autonomous agents are organized into "planets," each functioning as an isolated problem-solving unit. Within a planet, agents tackle sub-problems independently, contributing to distributed task management and reducing computational bottlenecks.

The planets operate under the coordination of a central hub, which facilitates interplanetary communication and integrates the outputs into cohesive solutions. This decentralized architecture ensures modularity, enabling the system to handle complex tasks while maintaining scalability and adaptability.

3.2 Iterative Refinement and Debate

A defining feature of the framework is the iterative refinement process, inspired by principles of collaborative reasoning. Agents within a planet engage in structured debates, proposing solutions, critically evaluating peer responses, and revising their outputs. Each debate round enhances the quality of solutions, guided by semantic similarity metrics computed using Sentence Transformers [4].

This iterative mechanism continues until a convergence threshold is met, ensuring that outputs are both accurate and contextually aligned. The debate mechanism fosters diversity in agent perspectives while steering the group toward a consensus-driven resolution.

3.3 Framework Transition and Enhancements

Initially implemented using OpenAI Swarm, the system established foundational elements such as agent parallelization and response aggregation. Transitioning to Microsoft Autogen introduced several enhancements:

- **Dynamic Role Allocation:** Autogen allows real-time role reassignment for agents, enhancing the framework's ability to adapt dynamically to evolving problem requirements.
- **Society of Mind Wrapper:** This feature supports nested group communication, enabling agents across different planets to interact and collaboratively refine their contributions. Such interactions expand the scope of problem-solving beyond individual planets.
- **Semantic Convergence Metrics:** By integrating Sentence Transformers, we quantitatively assess the alignment of agent responses, ensuring that solutions are consistent and semantically coherent.

3.4 Agent Specialization

To maximize efficiency, agents are assigned specialized roles:

- **Prompt Generators:** Create structured problem statements and instructions for agents to follow.
- **Checker Agents:** Evaluate the semantic similarity of responses to determine convergence.
- **Aggregator Agents:** Summarize planetary outputs into a cohesive response for interplanetary collaboration.

3.5 Evolutionary Strategies

To enhance performance further, we incorporated evolutionary strategies inspired by PromptBreeder [5]. Prompts are treated as individuals in a population, with their effectiveness evaluated through a fitness function that measures the convergence of agents' responses. The prompt evolution process involves the following steps:

- Initialization: An initial prompt is generated for each question, outlining instructions for agents to collaboratively solve problems.
- Fitness Evaluation: The fitness of a prompt is determined based on the average similarity of agents' responses, where higher similarity indicates better alignment and clarity of instructions.
- Mutation and Selection: New prompts are generated by mutating the best-performing prompt using predefined mutation prompts (e.g., rephrasing or simplifying) and thinking styles (e.g., critical or creative thinking). The prompt with the highest fitness is selected as the best candidate for subsequent use.
- Iterative Refinement: This process repeats across multiple generations, gradually improving the clarity and utility of prompts.

4 Implementation

4.1 Initial Framework: OpenAI Swarm

The project initially utilized OpenAI Swarm to establish a foundational multi-agent system. Each agent in this framework operated autonomously, generating responses to shared queries. These responses were aggregated and evaluated by dedicated agents, which identified the most consistent and contextually relevant output. This structure allowed for parallel processing of agent interactions, showcasing the potential of swarm intelligence in improving LLM performance.

However, while Swarm provided an effective baseline, its limitations became apparent as the project evolved. The framework lacked flexibility in task allocation and inter-agent communication, particularly for complex, adaptive problem-solving scenarios. These shortcomings necessitated the exploration of a more advanced framework capable of dynamic reconfiguration and improved collaborative processes. [1].

4.2 Enhanced Framework: Microsoft Autogen

The migration to Microsoft Autogen marked a significant step forward in the project, introducing a range of enhancements that addressed the constraints of the Swarm framework [2]. Key advancements include::

- **Planetary Topology:** Autogen employs a hierarchical structure where agents are grouped into planets, each functioning as an independent ecosystem of tasks. These planets tackle specific sub-problems in isolation, reducing interdependencies and improving scalability. A central hub oversees interplanetary communication, enabling the integration of planetary outputs into cohesive solutions.
- **Iterative Debate Mechanism:** Agents within each planet engage in structured debates, iteratively refining their responses based on peer feedback. Checker agents evaluate these responses using semantic alignment metrics, ensuring consistency and alignment with the intended solution.
- **Dynamic Communication:** Leveraging Autogen's "Society of Mind" wrapper, agents can dynamically reassign roles and participate in nested group interactions. This capability enhances the framework's adaptability, allowing agents from different planets to collaborate when addressing tasks.
- **Consensus Evaluation:** Aggregator agents synthesize planetary outputs into a unified response, which is then refined by the central hub. The integration of Sentence Transformers ensures that the final response meets a high semantic similarity threshold, maintaining both accuracy and coherence.

4.3 Progress in Codebase Migration

The migration from Swarm to Autogen is currently underway, with significant strides already achieved. The Swarm codebase, now fully developed, serves as a comprehensive reference for feature transfer. Key features such as iterative debates, semantic evaluation, and aggregator workflows have been successfully adapted to the Autogen environment. However, certain aspects of the Autogen codebase remain under development.

5 Example Workflow

The workflow of our multi-agent planetary framework is structured as follows:

- **Problem Initialization:** The process begins with a user query, which is analyzed and structured by a prompt generator agent. This agent categorizes the query, identifies its key components, and formulates a detailed instruction set tailored for the agents on each planet. The prompt generator ensures that the problem is effectively framed.
- **Planetary Debate:** Within each planet, agents independently generate solutions to the assigned sub-problem. These initial outputs undergo evaluation by checker agents, which compute semantic similarity using Sentence Transformers. If the responses lack sufficient convergence (e.g., a cosine similarity threshold of 0.90 is not met), agents engage in additional rounds of debate. This iterative process enables agents to refine their solutions collaboratively, improving both precision and contextual relevance.
- **Interplanetary Collaboration:** Once each planet achieves internal consensus, their outputs are aggregated by a central hub. This stage involves interplanetary collaboration, where delegates - one for each planet - vote on the final answer that will be provided to the user. If no majority vote is achieved among the delegates, a new generation of prompts is initiated and planets begin debating again on mutated prompts. Autogen's Society of Mind wrapper facilitates dynamic group interactions, allowing agents from different planets to collectively address overlapping or interdependent aspects of the problem.
- **Evolutionary Prompting:** When no majority vote is achieved in the Interplanetary Committee, the system leverages evolutionary prompting. The fitness of prompts is evaluated based on the degree of response convergence, guiding the generation of new, optimized prompts through mutation (e.g., rephrasing or simplification).
- **Final Output:** The central hub synthesizes the final response by combining the refined planetary outputs based on the majority vote.

6 Discussion

The swarm-based multi-agent system demonstrated clear advantages over single-agent approaches:

- **Accuracy:** Multi-agent collaboration consistently improves solution accuracy, particularly for complex queries. The iterative refinement process ensures that agents' collective outputs surpass the quality of individual contributions.
- **Scalability:** The planetary model enables distributed task management, reducing computational bottlenecks and allowing the system to handle increasingly complex problems. By

isolating sub-problems within planets, the framework optimizes resource allocation while maintaining overall coherence.

- **Flexibility:** Autogen's dynamic role allocation and nested group communication capabilities significantly enhance the adaptability of the system. Agents can easily reconfigure their roles and interact across planetary boundaries, ensuring efficient problem-solving.

Despite these achievements, challenges remain:

- **Computational Overhead:** The iterative debate and interplanetary collaboration mechanisms, while effective, still increase computational demands, particularly for large-scale systems. Optimizing these processes is essential to maintaining efficiency without compromising performance.
- **Convergence Metrics:** The current reliance on semantic similarity thresholds to determine debate convergence may not always yield optimal results. While effective for initial experiments, this metric requires further validation to ensure its robustness across diverse problem domains. Alternative convergence measures, such as task-specific fitness functions, may offer improvements.
- **Framework Transition:** The migration from Swarm to Autogen introduces complexities in transferring features and ensuring compatibility. While the phased approach has minimized disruptions, continued refinement is necessary to fully leverage Autogen's capabilities.

7 Conclusion

This project highlights the potential of swarm intelligence in augmenting the performance and adaptability of large language models (LLMs). By integrating decentralized multi-agent collaboration with iterative refinement, the planetary framework addresses key limitations of standalone LLMs, improving accuracy, contextual understanding, and problem-solving capabilities. The phased migration from OpenAI Swarm to Microsoft Autogen has further enhanced the system's flexibility, scalability, and dynamic role allocation, paving the way for more robust multi-agent AI systems.

While the current implementation demonstrates significant advancements, challenges such as computational overhead, convergence metric optimization, and the migration to the Autogen framwork. Addressing these challenges will be critical in unlocking the full potential of the framework. Future work will prioritize developing hierarchical agent architectures, refining prompt evolution strategies, and leveraging distributed processing to scale the system effectively.

References

[1] OpenAI. Swarm framework documentation, 2023.

[2] Microsoft. Autogen documentation, 2023.

[3] Marvin Minsky. *The Society of Mind*. Simon and Schuster, 1986.

[4] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[5] Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2305.18354*, 2023.