Smart Media Journal / Vol.14, No.10 / ISSN:2287-1322
https://dx.doi.org/10.30693/SMJ.2025.14.10.140

거대 언어 모델 기반 멀티에이전트 토론 시스템의 상호작용 아키텍처별 성능 비교

(Multi-Agent Debate System based on Large Language Model: Comparative Analysis of Interaction Architectures)

임보정*, 서호건**
(Bojeong Im, Hogeon Seo)

요 약

본 논문은 대규모 언어 모델(Large Language Model: LLM) 기반 멀티에이전트 시스템에서 시스템의 아키텍처가 토론의 품질에 미치는 영향을 분석한다. 이를 위해 GPT-4o, Gemini-2.5-Pro, Gemini-2.5-Flash, Gemini-2.0-Flash, Llama-4-maverick, Deepseek-chat-v3-0324, Claude-Sonnet-4의 7가지 LLM를 이용하여 토론 논제를 평가했다. 그리고 각 모델의 응답 성능을 비교한 후, 본 논문에서 이용할 대표 언어 모델로 Gemini-2.5-Flash를 선정했다. 멀티에이전트 상호작용 방식으로 MagenticOne, Swarm, SelectorGroupChat, RoundRobinGroupChat의 네 가지 토론 아키텍처를 이용하여 토론 시스템을 구축하였으며, 대표 모델 선정 과정에서 점수가 높은 상위 10개의 논제를 기반으로 토론을 진행하였다. 실험 결과로는 대화의 흐름에 따라 발언자를 동적으로 선택하는 SelectorGroupChat 아키텍처가 토론 점수에서 가장 높은 점수를 받았다. 본 논문은 멀티에이전트 기반 시스템을 활용하여 토론에 가장 적합한 아키텍처를 확인하였다.

■ 중심어 : 토론 시스템 ; 상호작용 아키텍처 ; 선택기 그룹 채팅 ; 멀티에이전트 시스템 ; 대규모 언어 모델

Abstract

This study analyzes the impact of system architecture on the quality of debates in multi-agent systems based on large language models (LLMs). To this end, debates were evaluated using seven LLMs: GPT-4o, Gemini-2.5-Pro, Gemini-2.5-Flash, Gemini-2.0-Flash, Llama-4-maverick, Deepseek-chat-v3-0324, and Claude-Sonnet-4. After comparing the response performance of each model, Gemini-2.5-Flash was selected as the representative LLM for this research. A multi-agent debate system was then constructed using four architectures—MagenticOne, Swarm, SelectorGroupChat, and RoundRobinGroupChat—and debates were conducted based on the top 10 topics with the highest scores from the model selection process. The experimental results showed that the SelectorGroupChat architecture, which dynamically selects speakers according to the flow of conversation, achieved the highest debate scores. This study identifies the most suitable architecture for debates utilizing a multi-agent-based system.

■ keywords : Debate System ; Interaction Architecture ; Selector Group Chat ; Multi-agent System ; Large Language Model

Ⅰ. 서 론

최근 거대 언어 모델(Large Language Model: LLM)의 발전으로 LLM 기반의 인공지능 시스

* 준회원, 한국원자력연구원 인공지능응용연구실 인턴연구원
** 정회원, 과학기술연합대학원대학교 인공지능전공 부교수, 한국원자력연구원 인공지능응용연구실 선임연구원
이 연구는 한국원자력연구원 기본연구개발사업 (KAERI-524540-25), 대한민국 정부 (과학기술정보통신부)의 재원으로 이뤄진 국가과학기술연구회 글로벌 TOP 전략연구단 지원사업(No. GTL24031-000), 한국연구재단 기본연구사업(RS-2023-00253853), 정보통신산업진흥원 고성능컴퓨팅 지원 사업의 지원을 받아 수행됨.
접수일자 : 2025년 08월 19일
수정일자 : 2025년 09월 25일
게재확정일 : 2025년 10월 05일
교신저자 : 서호건 e-mail : hogeony@hogeony.com

템이 단순한 질의응답을 넘어 집단 토론, 분산 의사결정 등 멀티에이전트 상호작용 영역으로 그 활용 범위를 넓혀가고 있다[1-3]. 특히, LLM 기반 멀티에이전트 토론 시스템은 사회적으로 다양한 영역에서 활용될 수 있는 잠재력을 지니고 있다. 그러나 복수의 에이전트 구성 방식이나 상호작용 아키텍처의 선택에 따라 토론 결과의 질적 차이가 발생할 수 있다[4]. 그러나 이에 대한 체계적이고 실증적인 비교 연구는 충분히 이루어지지 않았다.

멀티에이전트 토론 시스템은 멀티에이전트 상호작용 연구에서 에이전트 간의 효과적인 논의와 최적의 합의점에 도달하는 것에 중요한 역할을 할 것이다. 하지만, 발언 순서나 제어 방식을 어떻게 설계하는지에 따라 토론 품질이 크게 좌우된다는 한계도 존재한다. 예를 들어, 사회자(Moderator)의 유무, 발언 순서의 고정이나 동적 조정 등 다양한 요소로 인하여 토론의 과정과 결과가 크게 달라질 수 있다.

본 논문은 LLM 기반 멀티에이전트 토론 시스템에서 아키텍처가 토론의 품질에 미치는 영향을 체계적으로 분석한다. 이를 위해, 첫 번째로 Gemini-2.5-Flash, Gemini-2.5-pro, Gemini-2.0-Flash, GPT-4o, Claude-Sonnet-4, Deepseek-chat-v3-0324, Llama-4-maverick 일곱가지 LLM을 대상으로 토론 주제 평가를 수행하였다. 그 결과 7가지 LLM이 평가한 점수의 중앙값에 가장 근접한 점수를 준 모델(Gemini-2.5-Flash)을 대표 모델로 선정하였다. AutoGen [7]의 RoundRobinGroupChat, Swarm, SelectorGroupChat, MagenticOne으로 네 가지 토론 아키텍처를 활용하여 토론 시스템을 개발하고, 시스템별로 10개 주제에 대해 3회씩 토론을 진행하였다. 생성된 토론 대본은 토론 대본 평가 시스템을 적용하여 논증 구조, 상호작용, 숙의성 등 다양한 관점에서 비교·분석하였다. 실험 결과, 발언 타이밍과 흐름을 유연하게 제어하는 SelectorGroupChat 아키텍처가 논증적 사고와 숙의 과정에서 가장 뛰어난 성능을 보였다.

본 논문의 구성은 다음과 같다. 제2장에서는 LLM 기반 멀티에이전트 및 토론 시스템 관련 선행 연구를 살펴보고, 제3장에서는 실험에 대한 전체적인 과정을 설명한다. 그리고 제4장에서는 실험 결과를 다루고 있고, 마지막으로 제5장에서는 결론을 제시한다.

Ⅱ. 관련 연구

현재 LLM 기반 멀티에이전트로 에이전트의 구성, 토론 진행 방식 등 다양한 상호작용 방법을 연구하고 있다. 한 가지 LLM로 구성된 멀티에이전트 보다 여러 개의 서로 다른 LLM을 활용한 멀티에이전트로 협업하였을 때 더욱 산수 문제를 잘 푸는 등 정답률이 이전에 비해 훨씬 오른 것을 알 수 있다[1]. 그리고 여러 에이전트가 각자 찬성과 반대 견해로 토론을 진행하고, 판사가 이를 평가하여 최종 답을 정하는 방식을 통해 답변의 다양성과 정확도를 높일 수 있음을 보여주었다[5]. 판사와 같이 제3의 LLM을 중재자로 두고 에이전트 간 논의 과정에서 발생하는 환각을 해결하여 응답의 신뢰성과 일관성을 높이는 방법도 제안되었다[6]. 그러나 에이전트 수와 토론 횟수가 늘수록 비용이 크게 증가한다는 한계가 있어, 이를 위해 에이전트들을 여러 그룹으로 나눈 후 그룹의 내부와 외부에서 토론을 진행하고 결과를 공유하는 과정을 반복하는 GroupDebate 구조를 제안하여 토론 시스템의 효율성과 성능을 모두 향상토록 하였다[7].

이러한 연구에 이어 LLM 기반 멀티에이전트를 이용한 대화 프레임워크인 AutoGen이 제안되었다[8]. 본 논문에서는 기존 연구들이 제안한 에이전트 수, 토론 라운드 등의 요소 외에 AutoGen 프레임워크를 활용하여 실제 토론 상황에 적합한 멀티에이전트 시스템 구조는 무엇인지에 초점을 맞추었다. 이를 위해 AutoGen이 제공하는 4가지(Swarm, MagenticOne, SelectorGroupChat, RoundRobinGroupChat) 의

아키텍처를 적용하여 멀티에이전트 토론 시스템을 구축하였다. 그리고 어떤 아키텍처가 토론 환경에 가장 적합한지 비교 분석하였다.

Ⅲ. 실험 설계

1. 연구 실험 개요

본 논문은 멀티에이전트 시스템 구조 중 토론에 가장 적합한 것을 찾기 위해, 세 단계로 나누어 실험을 진행했다. 첫 번째 실험으로는 연구에 사용할 대표 모델을 선정하였다. 이를 위해 GPT-4o, Gemini-2.5-Pro, Gemini-2.5-Flash, Gemini-2.0-Flash, Llama-4-maverick, Deepseek-chat-v3-0324, Claude-Sonnet-4의 7가지 LLM을 이용하여 토론 논제 평가자 에이전트를 구축하였다. 토론의 성과를 객관적으로 비교하기 위해 토론의 논제와 관련된 논문을 참고하여 토론의 질을 정량적으로 평가할 수 있는 평가표를 구축하였고, 이를 표 1에 제시하였다[8]. 이후 평가자들에게 해당 토론 논제 평가표를 제시하여 20개의 토론 주제에 대해 각 모델의 응답을 평가하도록 지시하였다. 평가된 각 LLM의 점수를 산출한 뒤, 모델별 중앙값에 가장 가까운 값을 보인 'Gemini-2.5-Flash' 모델을 대표 모델로 선정하였다.

두 번째 단계에서는 AutoGen[7]을 기반으로 한 네 가지 멀티에이전트 아키텍처(Swarm, RoundRobinGroupChat, SelectorGroupChat, MagenticOne)를 이용하여 멀티에이전트 토론 시스템을 구축하였다. 그리고 첫 번째 단계에서 선정된 10개의 토론 주제에 대해 3회씩 토론을 진행하였다.

표 1. 토론 논제 평가표
Table 1. Debate Topic Rubric

Evaluation Items	Max Score
1. Balance	5
2. Equivalence of Arguments	5
3. Timeliness and Social Significance	5
4. Educational Effect and Interest	5
5. Ease of Data Collection	5
6. Appropriateness of Scope	5
7. Clarity of Terms and Sentences	5
8. Policy Orientation	5
Total	40

세 번째 단계에서는 각 시스템이 생성한 총 30회의 토론 결과를, 최종 토론 평가자 에이전트를 이용하여 정량적 및 정성적으로 분석·평가하였다. 이를 통해 다양한 에이전트 시스템 아키텍처 중 LLM 기반 토론에 가장 적합한 방식을 규명하고자 하였다.

2. 모델 선정 과정

본 논문에서는 실험에 사용할 대표 LLM 모델을 선정하기 위하여 다음과 같은 절차를 거쳤다. 먼저, 자체적으로 작성한 토론 주제 평가표(총 8개 항목, 각 항목 5점 만점, 총 40점 만점)를 바탕으로 평가를 진행하였다. 평가 항목은 균형성, 논거의 대등함, 시의성 및 사회적 중요성, 교육적 효과 및 흥미, 자료 조사의 용이성, 범위의 적절성, 용어 및 문장의 명확성, 정책 지향성으로 구성되었다[9].

평가에 쓴 모델은 GPT-4o, Gemini-2.5-Pro, Gemini-2.5-Flash, Gemini-2.0-Flash, Llama-4-maverick, Deepseek-chat-v3-0324, Claude-Sonnet-4의 7가지 주요 모델이다. 각 LLM 모델별로 총 20개의 토론 주제를 평가표 기준에 따라 개별 점수를 매겼다. 이후, 각 모델별 전체 주제에서 획득한 점수의 평균을 산출하였다. 대표 LLM 모델 선정은, 산출된 평균 점수 중 전체 중앙값에 가장 근접한(중앙값에 가장 가까운 점수를 얻은) 모델을 선정하는 방식으로 진

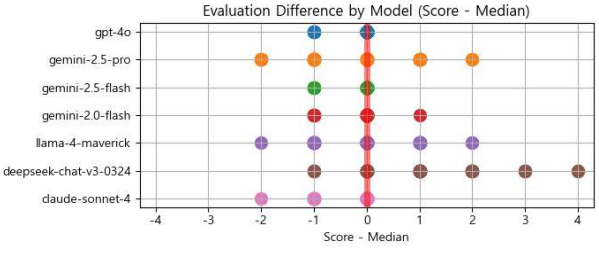


그림 1. 모델별 중앙값 대비 점수 차이
Fig 1. Score Difference from Median by Model

행하였다. 그림 1에서 확인될 수 있는 바와 같이 편차가 낮고 보다 최대 입력 토큰 길이는 더 길면서도 API 비용이 더 저렴한 Gemini-2.5-Flash를 본 논문의 모든 토론 에이전트 시스템 구현과 전반적인 실험에 기준 모델로 활용하였다. 이는 아키텍처 외의 변수의 영향을 최소화하여서, 다양한 시스템 아키텍처에 대해 일관된 비교가 가능하도록 하였다.

3. 토론 아키텍처 설계

가. RoundRobinGroupChat

RoundRobinGroupChat 아키텍처는 그림 2의 (a)와 같이 순환 방식으로 대화를 주고받는 구조이다[10]. 사회자(Moderator)와 네 명의 토론자(찬성 2명, 반대 2명)로 구성된 순차적으로 진행되는 토론 시스템이다. 이 구조의 핵심은 정해진 순서대로 각 참가자에게 발언권이 순환적으로 부여되는 점에 있다. 즉, 사회자로부터 시작하여 사전에 정해진 라운드로빈(RoundRobin) 순서(사회자 → 찬성1 → 반대1 → 찬성2 → 반대2 → ...)에 따라 각 토론자가 순서대로 발언한다.

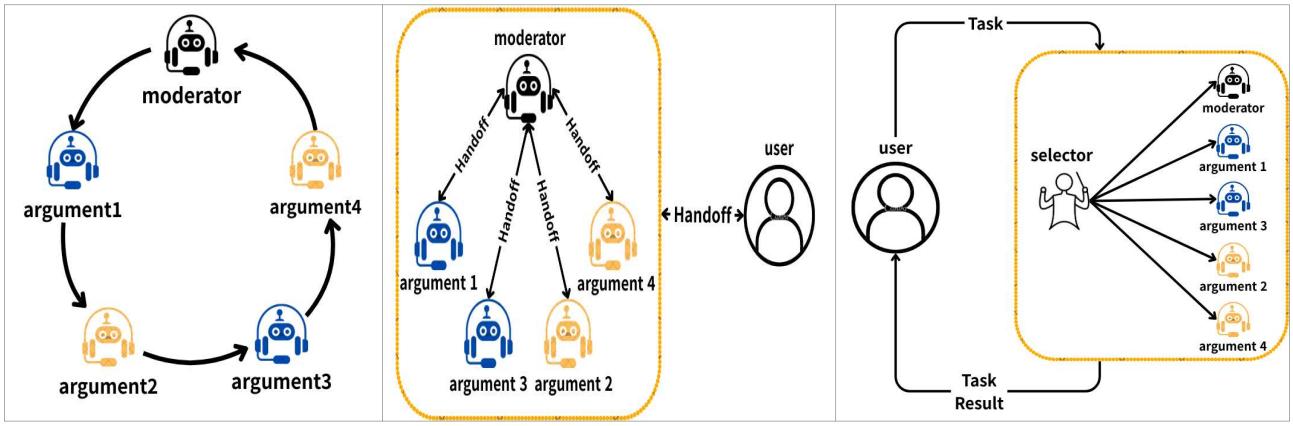
토론의 흐름은 크게 입론, 질의·답변, 반박, 최종 주장 등의 단계로 나뉘며, 사회자는 각 단계의 시작과 종료, 발언자 지정, 토론 진입 및 종료 선언의 객관적·공정한 관리 역할을 수행한다. 모든 토론자는 본인의 순서가 되었을 때에만 발언할 수 있고, 반드시 시스템적으로 요구되는 종료 문장(예: "발언을 마칩니다.")으로 발언을 끝내야 한다. 사회자만이 토론의 개시와 종료를 선언할 수 있다.

이 구조에서는 발언권 위임이 미리 정해진 순서대로 자동적으로 순환되기 때문에, 실험자는 복잡한 발언 순서 지정이나 중재 절차 없이 간단하게 복수 에이전트의 토론 실험을 반복 수행할 수 있다. 단, 토론자가 논의 흐름 또는 규칙에서 벗어나거나, 시스템적으로 의미 없는 발언을 반복하는 경우 사회자가 중재 및 발언권 예외 제어를 수행하도록 설정하였다.

나. Swarm

Swarm 아키텍처는 그림 2의 (b)와 같이 다수의 에이전트가 협력하며 상호작용하는 분산형 구조이다. 사회자(Moderator) 1명과 찬성과 반대 각 2명씩 구성되어 4명의 토론자 에이전트가 참여하는 구조이다[11]. 본 아키텍처의 가장 큰 특징은 위임(Handoffs) 기능이다. Handoffs는 발언권과 같은 권한을 에이전트 간에 명시적으로 전달할 수 있도록 하는 기능이다.

Swarm에서는 발언권의 관리를 사회자가 담당하도록 하였다. 각 토론자 에이전트는 자신의



(a) RoundRobinGroupChat (b) Swarm (c) SelectorGroupChat

그림 2. AutoGen 기반 멀티에이전트 토론 시스템의 아키텍처(RoundRobinGroupChat, Swarm, SelectorGroupChat)
Fig 2. Architectures of an AutoGen-Based Multi-Agent Debate System (RoundRobinGroupChat, Swarm, SelectorGroupChat)

발언이 끝났을 때마다 반드시 “발언을 마칩니다.”라는 문장을 한 뒤 Handoffs 기능을 통해 발언권을 사회자에게 넘기고 종료하도록 하였다.

토론자들의 발언권을 받은 사회자는 토론 흐름과 상황을 고려하여 다음 발언자를 직접 지정하고, 해당 에이전트에게 마찬가지로 Handoffs 기능을 이용하여 발언권을 넘겨주며 토론의 흐름을 관리한다.

사회자는 발언권 관리 외에도 논의가 과도하게 격해지거나 토론자의 발언이 반복되어 비생산적으로 흐를 경우에는 토론자들을 중재하고 토론이 원활하게 진행되도록 설정하였다. 토론의 종료 권한도 사회자만이 할 수 있도록 부여하여서 토론이 안정적으로 종료될 수 있도록 하였다.

Swarm 아키텍처는 이처럼 모든 발언과 발언권의 전달 과정이 사회자를 중심으로 통제함으로써, 중복 발언이나 주제를 이탈 행위 등의 문제를 방지하도록 설정하였다.

다. SelectorGroupChat

SelectorGroupChat 아키텍처는 그림 2의(c)와 같이 대화 중 특정 상황이나 조건에 따라 발언자를 동적으로 선택하는 구조이다[12]. 사회자(Moderator) 1인과 찬성·반대 각 2인으로 구성된 네 명의 토론자 에이전트가 참여하며, 발언 순서의 주도적 제어를 ‘Selector’(순서 선택 담당 로직)에 위임하는 것이 핵심 특징이다.

본 구조에서는 각 토론 단계(입론, 질의응답, 반박, 최종 주장 등) 및 발언이 끝날 때마다, Selector가 직전 대화의 맥락과 미리 정의된 토론 흐름 규칙을 바탕으로 가장 적절한 다음 발언자를 자동으로 선정한다. 대화가 시작되면 사회자가 토론을 개시하고, 각 토론자가 자신의 차례에만 발언을 수행하며, 발언이 종료되면 Selector가 현재 상황과 규칙에 맞추어 발언권을 다음 참가자에게 넘겨준다.

이때 사회자는 각 단계의 관리, 요약, 다음 발언자 지명 등 총괄적 진행을 담당하며, 모든 참가자는 규정된 종료 문구(예: “발언을 마칩니다.”)로 발언을 끝맺음으로써 절차적 일관성을 유지한다. 모든 토론 절차가 종료되면, 사회자가 “토론 종료하겠습니다.”라는 최종 발언을 통해 세션을 공식적으로 마친다. SelectorGroupChat은 RoundRobinGroupChat 구조의 고정된 순회 방식과 달리, 토론 맥락 및 상황을 반영한 유연한 발언권 제어와 적응적 절차 구성이 가능한 점이 주요 장점이다.

라. MagenticOne

MagenticOne 아키텍처는 그림 2의 (d)와 같이 오케스트레이터(MagenticOrchestrator)가 모든 토론의 흐름을 관리하는 중앙 집중형 다중 에이전트 구조이다[13]. 이 아키텍처는 네 명의 토론자(찬성 2명, 반대 2명)와 한 명의 오케스트레이터로 구성된 구조적 토론 시스템이다. 토론자들은 자신의 차례에 따라 발언을 하고, 발언 끝에 반드시 “발언을 마칩니다.”라는 종료 문장으로 끝날 수 있도록 지정하여서 대화의 흐름이 일관

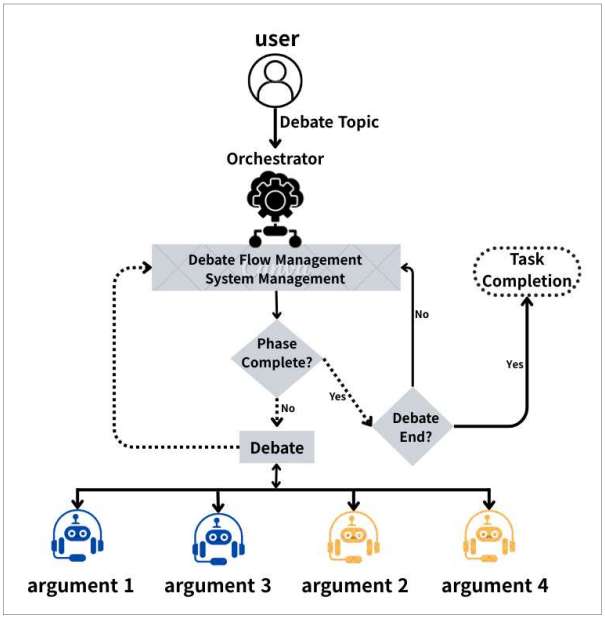


그림 3. AutoGen 기반 멀티에이전트 토론 시스템의 MagenticOne 아키텍처
Fig 3. MagenticOne Architectures of an AutoGen-based Multi-Agent Debate System

되게 이어지도록 하였다.

이 아키텍처의 가장 큰 특징은 오케스트레이터가 사회자(Moderator)의 모든 역할을 동시에 수행한다는 점이다. 실험 초기에는 사회자 에이전트와 오케스트레이터를 별도로 두고 오케스트레이터에게는 사회자의 역할을 침범하지 않고 보조만 하도록 설정하였지만, 두 에이전트의 역할과 권한이 중복되어 토론 진행 과정에서 두 에이전트 간의 발언이 중복되는 문제가 발생하였다. 이에 따라 사회자 에이전트는 제외하고, 오케스트레이터가 토론의 전 단계(입론, 자유토론, 최종 주장 등)를 전체적으로 관리하고, 발언 순서 지정 및 토론 종료 선언 등 사회자의 역할까지 모두 담당하도록 재설정하였다.

오케스트레이터는 영어로 발언이 되는 것이 기본 설정으로 되어 있어서 토론 진행에서는 자연스러운 한국어로 발언할 수 있도록 지시하였으며, 매 발언 시 마지막에 “다음 발언자: [에이전트 이름]”의 형식으로 다음 발언자를 명확하게 명시하여 발언 충돌을 방지하였다. 토론은 오케스트레이터의 지시로 진행되며, 오케스트레이터가 모든 발언 순서와 토론의 흐름을 관리한다. 각 에이전트는 오케스트레이터 안내에 따라 발언하도록 프롬프트를 설정하였다.

4. 토론 평가

가. 평가 지표

토론 최종 평가는 표 2와 같이 총 3개의 파트(Part A, B, C)로 구성되어있다. 각 파트는 3개의 하위 항목으로 세분화하고, 각 항목의 중요도에 따라 차등 부여하였다. 본 논문에서는 토론 참여자, 사회자, 전체 토론 과정을 균형 있게 평가할 수 있도록 평가표를 구축하였고, 이 평가표를 평가자 모델에게 제시하여 정량적 점수를 산출하였다[14].

나. 평가 방식

토론 결과물의 평가는 Gemini-2.5-Flash를 이용하여 최종 평가자 에이전트를 생성하여, 평가하도록 지시하였다. 평가자에게 토론 평가표를 지시하여 모든 실험에서 동일한 기준으로 토론을 평가하였다. 모델에게는 정해진 프롬프트를 입력하여, 표2에 제시된 평가표 항목에 따라 응답을 생성하도록 하였다.

다. 자동 평가 시스템

총 120개의 토론 결과를 객관적이고 일관성 있게 평가를 하기 위해, 자동화된 평가 시스템을 구축하였다. 시스템 구축은 3단계로 나뉜다.

(1) 실험 결과물(총 10가지 주제 x 4가지 토론 시스템 구조 x 3회 반복)은 각각 텍스트 파일로 저장하였으며, 이 파일들을 순차적으로 불러오도록 자동 평가 코드를 구현하였다.

(2) 각 토론 결과 텍스트는 사전 정의된 최종 평가자 에이전트에게 전달하고, 프롬프트에 제시된 평가표를 보고 평가 결과를 생성하였다.

(3) 응답으로 생성된 평가 보고서는 개별 텍스트 파일로 저장되어, 이후 네 가지 멀티에이전트 토론시스템 성능 비교에 활용된다.

표 2. 토론 평가표
Table 2. Debate Rubric

Evaluation Items	Score
Part A: Debate Participants	**(Total: 40)**
1. Argument Construction	15
2. Interaction	15
3. Debate Attitude	10
Part B: Moderator	**(Total: 30)**
1. Fairness & Neutrality	10
2. Process Management	10
3. Debate Facilitation	10
Part C: Debate Process & Interaction	**(Total: 30)**
1. Deliberation	10
2. Quality of Participation	10
3. Discourse Environment	10
Total Score	**100**

Ⅳ. 실험 결과

1. 실험 조건

본 논문에서 대표 LLM은 실험 1단계 결과로 비용과 성능을 고려하여 Gemini-2.5-Flash로 확정했다. 아키텍처 별 구축한 네 가지 토론 시스템의 차이는 크게 사회자의 유무와 발언자 배치 구조라로 할 수 있다. 시스템별 토론자 수는 4~5인으로 구성했다.

모든 토론은 동일 프롬프트 규칙을 적용하여 실험에 일관성을 부여하였다. 사회자 에이전트는 토론 시작, 단계별 안내, 발언권 배분, 흐름 통제 및 마무리 등 관리적 역할을 담당하며, 찬성·반대 토론자는 주장·반박·근거 제시, 그리고 반드시 “발언을 마칩니다.”로 발언을 종료하는 규칙을 준수하도록 하였다. 사회자의 호출 전에는 추가 발언이 금지된다.

토론 과정의 모든 발언 내역은 실시간으로 수집·저장되어, 발언자와 발언 내용이 텍스트 파일(txt) 형태로 기록된다. 실험 중 시스템 장애 또는 예외가 발생할 경우, 최근 대화 로그를 바탕으로 이어받기 기능을 통해 실험의 연속성이 유지된다.

2. 실험 결과

총 10개의 토론 주제를 대상으로 시스템별로 3회씩 토론을 반복하였다. 그 결과로 얻은 120개의 토론 대본을 평가하여, 어떤 시스템이 토론의 품질이 높은지 분석하였다. 그림 3은 각 시스템별로 10개의 주제에 대해 3회 평가 점수들의 평균을 시각화한 결과이고, 그림 4는 시스템별 토론 점수의 표준 편차를 나타낸 그래프이다.

Swarm은 그림 4를 보면 topic 2번에서 낮은 점수를 보였는데, 이는 입론 단계에서 토론이 종료되어 상호작용이 전혀 이루어지지 않았던 문제로 확인됐다. 이로 인해, 표 2의 Part A와 Part C에서 낮은 점수를 받았다.

RoundRobinGroupChat은 그림 4에서 topic 4에서 낮은 점수를 받은 것을 확인할 수 있다. 이는 토론자들의 논의가 발전되지 않고 반복적인 주장과 반박이 이어져 논의가 정체된 현상으로 일어난 결과이다. 이로 인해 표 2의 Part C의 상호작용 관련 항목에서 낮은 점수를 받았다. 토론자들의 논의가 정체되었을 때 사회자가 반복적인 발언을 인지하고 지적하였으나, 이러한 현상은 RoundRobinGroupChat의 특징 중의 하나인 발언 순서대로 상호작용이 이뤄지기에 사회자가 토론에 적극적으로 개입하기 어려웠다.

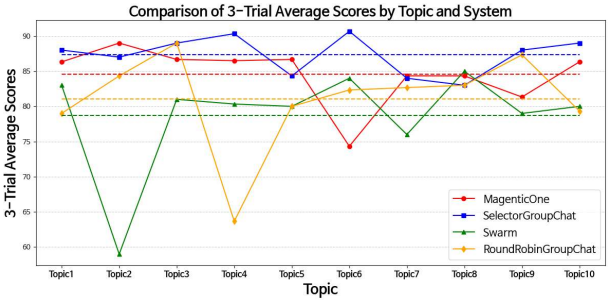


그림 4 각 주제에 대해 3회 시행한 점수의 평균
Fig 4. The Average Scores Across Three Trials by Topic

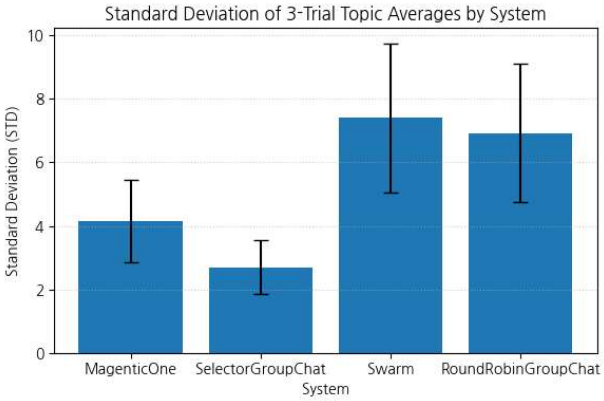


그림 5. 시스템별 점수 표준편차 및 오차 막대
Fig 5. Standard Deviation and Error Bar by System

표 3. 평균 점수 및 평균 점수의 표준편차
Table 3. Average Scores and Standard Deviations

System	Mean Score	Standard Deviation (Std)
MagenticOne	84.85	4.15
SelectorGroupChat	87.83	2.70
Swarm	78.83	7.40
RoundRobinGroupChat	81.37	6.93

MagenticOne은 초기 실험에서는 사회자 에이전트가 있었으나, MagenticOrchestrator와 서로 충돌이 발생하여 토론이 종료되거나, 같은 발언을 반복적으로 하는 등의 문제가 발생하였다. 이를 해결하고자 사회자 에이전트를 없애고 MagenticOrchestrator에게 사회자 권한을 부여하였으나, 몇몇 토론에서 MagenticOrchestrator는 발언자 지정만 하고, 사회자 역할인 발언자의 요약이나 심화된 논의를 할 수 있도록 유도하는 역할이 부족한 문제가 있었다. 그로 인하여 6번째 주제(Topic 6)에서 낮은 점수를 보였다.

그림 4을 보면 SelectorGroupChat은 다른 아키텍처보다 상대적으로 높은 평가점수를 받았고, 10개 주제 점수의 평균도 가장 높은 것으로 나타났다. 또한 SelectrGroupChat이 상대적으로 일관된 점수를 받은 것을 볼 수 있다. 그림 5에서 각 시스템의 표준편차를 살펴보면 MagenticOne은 4.15를, SelectorGroupChat은 2.70, Swarm은 7.40, RoundRobinGroupChat은 6.93으로 SelectorGroupChat 아키텍처가 표준편차가 가장 낮은 것으로 확인되었고, 안정적인 성능을 보이는 것을 확인할 수 있다. 네 가지 아키텍처 중 SelectorGroupChat이 토론 상황에서 상대적으로 가장 우수한 성능을 보인 것을 확인하였다.

Ⅵ. 결론

본 논문에서는 네 가지 멀티에이전트 아키텍처(Swarm, MagenticOne, SelectorGroupChat, RoundRobinGroupChat)를 이용한 다양한 멀티에이전트 토론 시스템을 구축하고, 실제로 어느 구조가 MAD(Multi-Agent Debate) 환경에서 가장 적합한지 분석하였다. 기존 연구들이 주로 에이전트 수, 토론 라운드, 판사 개입과 같은 요소에 초점을 맞추었지만, 본 논문은 멀티에이전트 시스템 구조 자체가 토론의 양질에 미치는 영향을 중점적으로 다루었다는 점에서 차별성을 가진다.

토론 논제 선정과 토론의 질을 객관적으로 평가하기 위해 구축한 정량적 평가표를 활용하였다. 그 후 Gemini-2.5-Flash 모델을 활용하여 각 아키텍처에서 생성된 토론 대본을 일관되게 평가했다. 결과로 SelectorGroupChat 아키텍처가 토론 시스템에서 가장 우수한 성능을 보였다. 이는 멀티에이전트 토론 시스템에서 단방향보다는 동적으로 발언권을 조절하는 구조가 멀티에이전트 간 상호작용 과정에서 더욱 효과적인 것을 알 수 있다.

이처럼 본 논문은 멀티에이전트 토론 시스템에 아키텍처가 토론의 품질에 영향을 미치는 것을 확인하였다. 평가에는 표 2의 평가표를 이용하여 LLM을 활용한 자동화 방식을 적용하였다. 하지만, 이러한 자동화된 평가 방식은 LLM이 상황에 따라 다르게 평가할 수 있다는 점에서 한계가 존재한다. 이를 보완하기 위해 정교하고 안정적인 평가 시스템을 위한 벤치마크 구축이 필요하다. 향후에는 본 아키텍처를 활용한 시스템에 다양한 모델 적용과 에이전트 간 반복적으로 토론을 진행하여 자기반성을 하는 과정을 통해 토론의 수준을 높이는 전략을 모색하고자 한다.

REFERENCES

[1] J. C. Y. Chen, S. Saha, M. Bansal, “Reconcile: Round-table conference improves reasoning via consensus among diverse LLMs,” arXiv preprint arXiv:2309.13007, Sep. 2023.

[2] M. Zhang, Z. Fang, T. Wang, S. Lu, X. Wang, T. Shi, “CCMA: A framework for cascading cooperative multi-agent in autonomous driving merging using Large Language Models,” *Expert Systems with Applications,* vol. 282, 127717, 5 Jul. 2025.

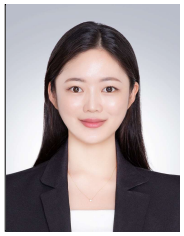
[3] 김윤민, 이재윤, 윤성민, “건물 운영 시그니처 분석을 위한 GPT 기반 Multi-Agent 모델 개발”, *한국생태환경건축학회 학술발표대회 논문집,* 제24권 제2호, pp. 72-73, 과학기술컨벤션센터, 대한민국, 2024년 11월

[4] B. Yan, Z. Zhou, L. Zhang, L. Zhang, Z. Zhou, D. Miao, Z. Li, C. Li, X. Zhang, “Beyond self-talk: A communication-centric survey of LLM-based multi-agent systems,” arXiv preprint arXiv:2502.14321, Feb. 2025.

[5] Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y.,

Wang, R., ... Tu, Z., “Encouraging divergent thinking in large language models through multi-agent debate,” arXiv preprint arXiv:2305.19118, May 2023.
[6] Z. Duan, J. Wang, “Enhancing multi-agent consensus through third-party LLM integration: Analyzing uncertainty and mitigating hallucinations in large language models,” *in Proc. of the 2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE),* pp. 2222 - 2227, Beijing, China, March 2025.
[7] T. Liu, X. Wang, W. Huang, W. Xu, Y. Zeng, L. Jiang, H. Yang, J. Li, “GroupDebate: Enhancing the efficiency of multi-agent debate using group discussion,” arXiv preprint arXiv:2409.14051, Sep. 2024.
[8] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, A. H. Awadallah, R. W. White, D. Burger, C. Wang, “AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation,“ arXiv preprint arXir:2308.08155, Aug. 2023.
[9] 박재현, “중학교 국어 교과서 토론 단원의 정책 주장 타당성 분석”, *새국어교육,* 제96호, pp. 139 - 165, 2013년
[10] AutoGen_Agentchat.Teams(2024). https://microsoft.github.io/autogen/stable/reference/python/autogen_agentchat.teams.html#autogen_agentchat.teams.RoundRobinGroupChat (accessed Jul., 8, 2025).
[11] Swarm(2024). https://microsoft.github.io/autogen/stable/user-guide/agentchat-user-guide/swarm.html. (accessed Jul., 8, 2025).
[12] SelectorGroupChat(2024). https://microsoft.github.io/autogen/stable/user-guide/agentchat-user-guide/selector-group-chat.html (accessed Jul., 8, 2025).
[13] Magentic-One(2024). https://microsoft.github.io/autogen/stable/user-guide/agentchat-user-guide/magentic-one.html (accessed Jul., 8, 2025).
[14] N. Sternlicht, A. Gera, R. Bar-Haim, T. Hope, N. Slonim, “Debatable Intelligence: Benchmarking LLM Judges via Debate Speech Evaluation,” arXiv preprint arXiv:2506.05062, Jun. 2025.

저 자 소 개



임보정(준회원)

2024년 동덕여자대학교 정보통계학과 학사 졸업.
2025년 한국원자력연구원 인공지능응용연구실 인턴연구원~

<주관심분야 : 인공지능, Agentic AI System, 생성형 인공지능>

서호건(정회원)

2013년 한양대학교 기계공학부 학사 졸업.
2018년 한양대학교 융합기계공학과 박사 졸업.

<주관심분야 : Agentic AI System, Multi-modal Data Reasoning AI, Autonomous NDT&E>