동계 세미나 (1/16)

AI Agent

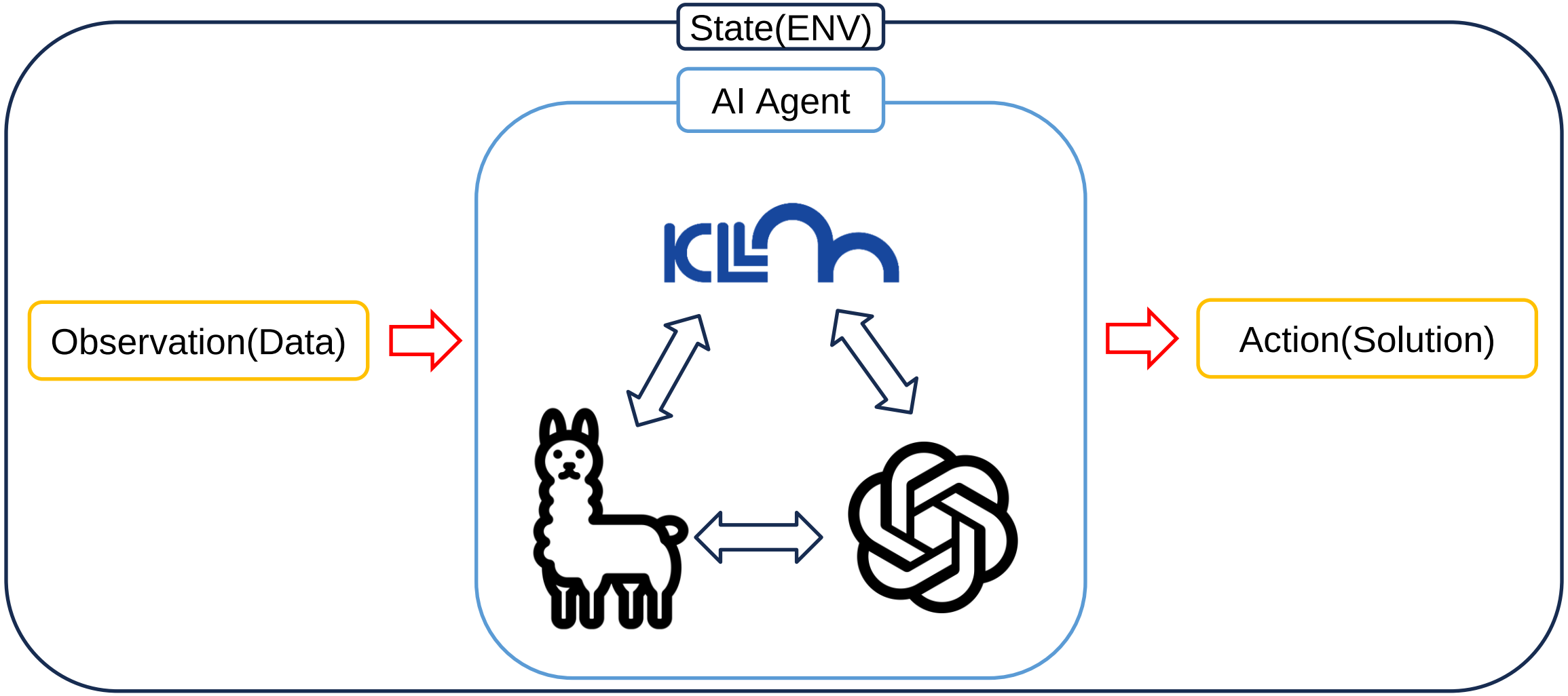
윤정호

Natural Language Processing & Artificial Intelligence

고려대학교 KOREA UNIVERSITY

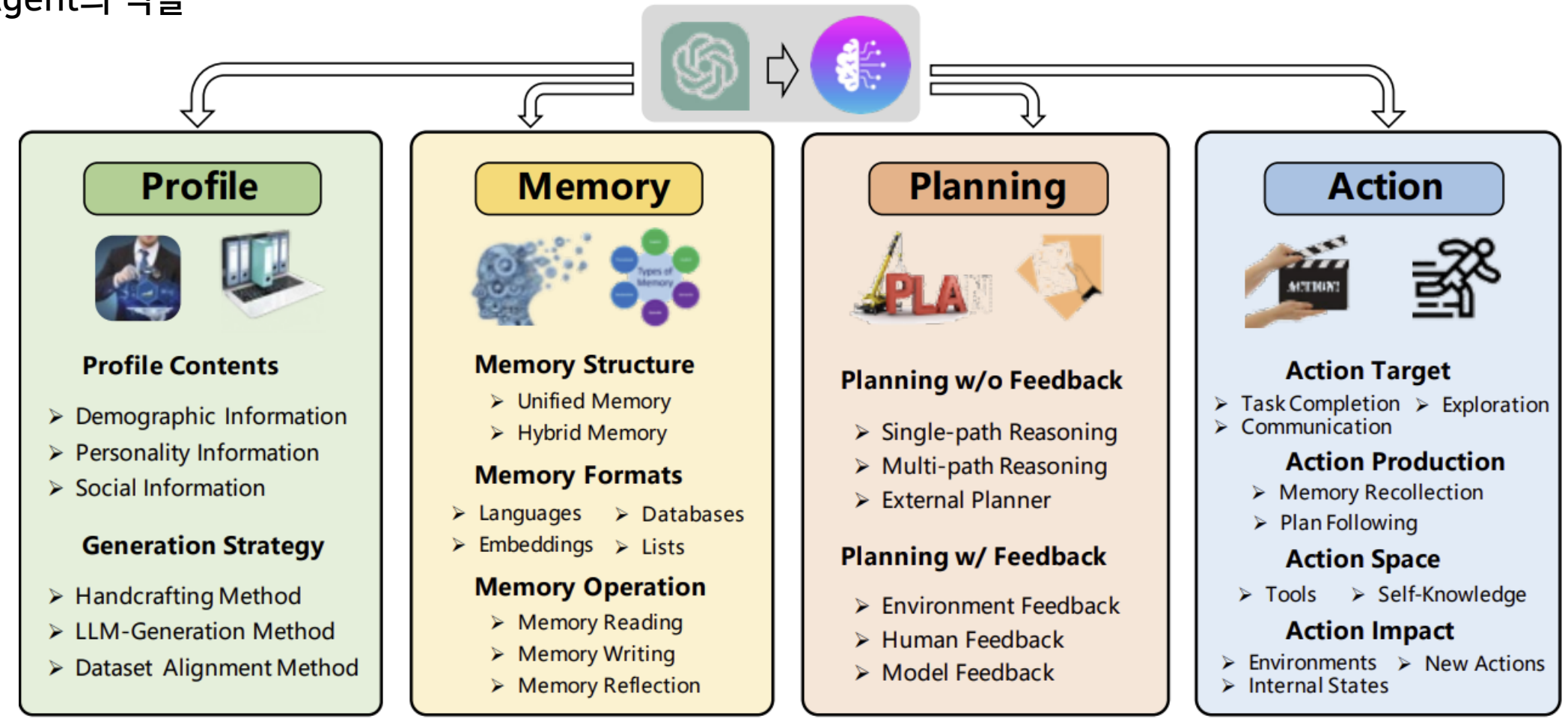
What is AI Agent

*인공지능(LLM)을 활용하여, 인간처럼 목표를 설정, 문제를 해결하며, 의사 결정을 내리는 지능형 시스템



What is AI Agent

*AI Agent의 역할



https://link.springer.com/article/10.1007/s11704-024-40231-1

ExpeL: LLM Agents Are Experiential Learners

Andrew Zhao,[♠] Daniel Huang, [♣] Quentin Xu, [♣] Matthieu Lin, [♣] Yong-Jin Liu, [♣] Gao Huang [♠*]

[♠] Department of Automation, BNRist, Tsinghua University
[♣] Department of Computer Science, BNRist, Tsinghua University
{zqc21,huang-jy22,xgd22,lyh21}@mails.tsinghua.edu.cn,
{liuyongjin,gaohuang}@tsinghua.edu.cn

AAAI 2024

Abstract & Introduction

특정 Task에 맞도록 모델을 fine-tuning하는 것은 리소스도 많이 들고 일반화 기능을 저하시키며 GPT, Claude와 같은 최신 모델에선 어려움

Parameter 업데이트 없이 경험에서 학습할 수 있는 새로운 방법론에 대한 필요성 대두

A computer program is said to **learn** from **experience** E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with **experience** E.

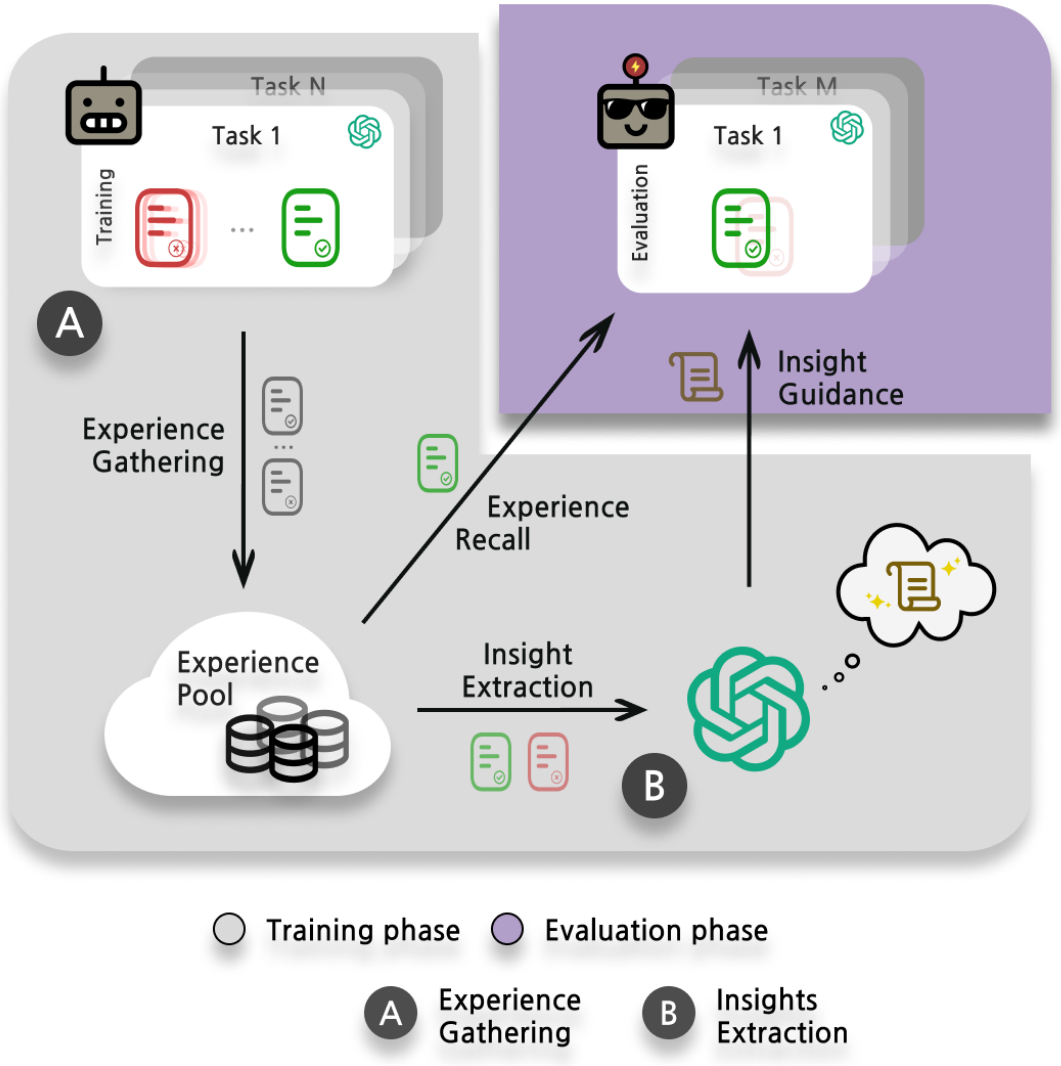
Tom Mitchell

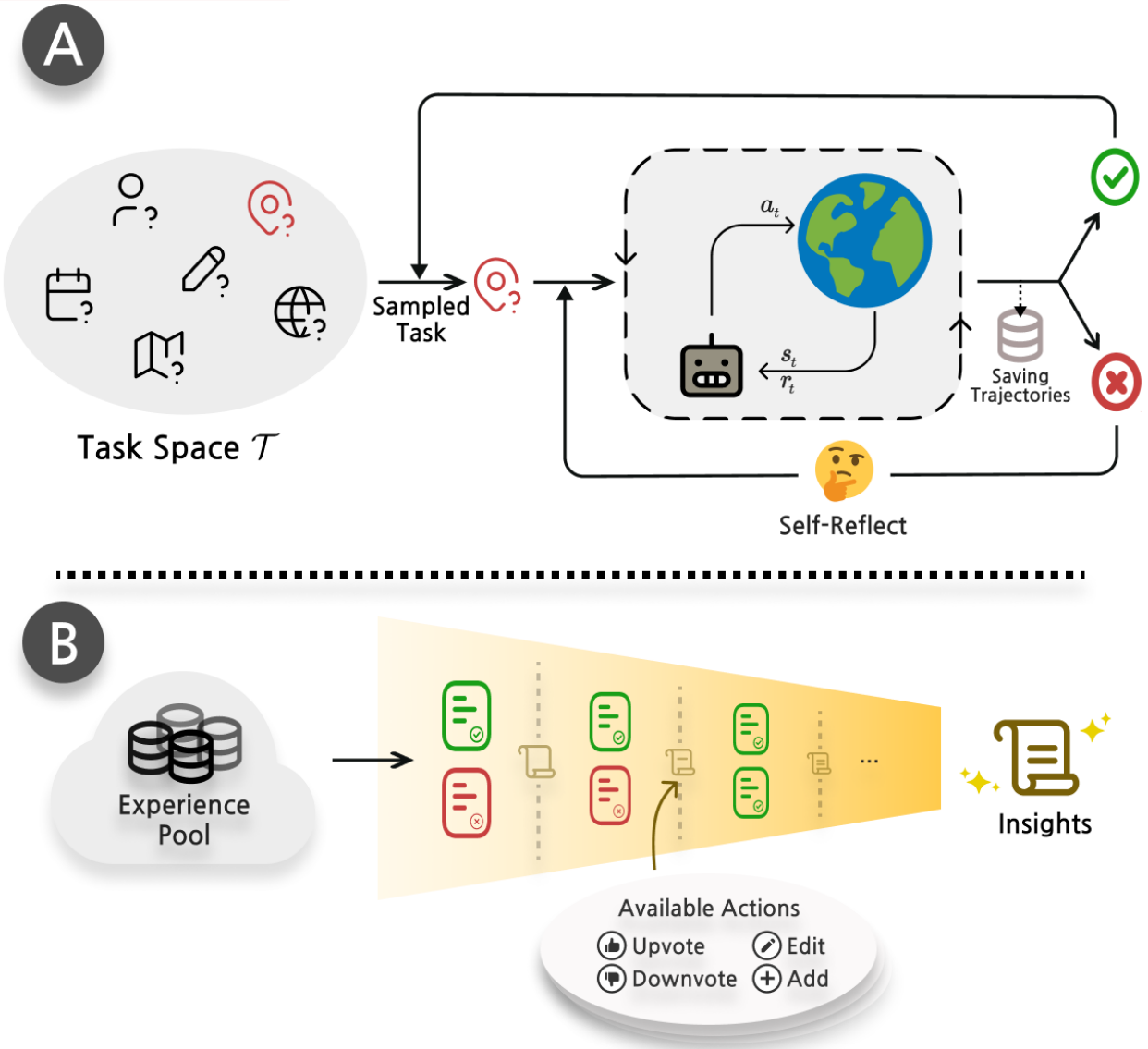
Abstract & Introduction

Few-Shot으로 성능 향상을 기대할 수 있지만 Context-Length의 한계 존재

시행착오를 통해 경험을 수집하고 Insight를 추출
성공적인 경험을 Retrieval하여
Few-Shot으로 사용

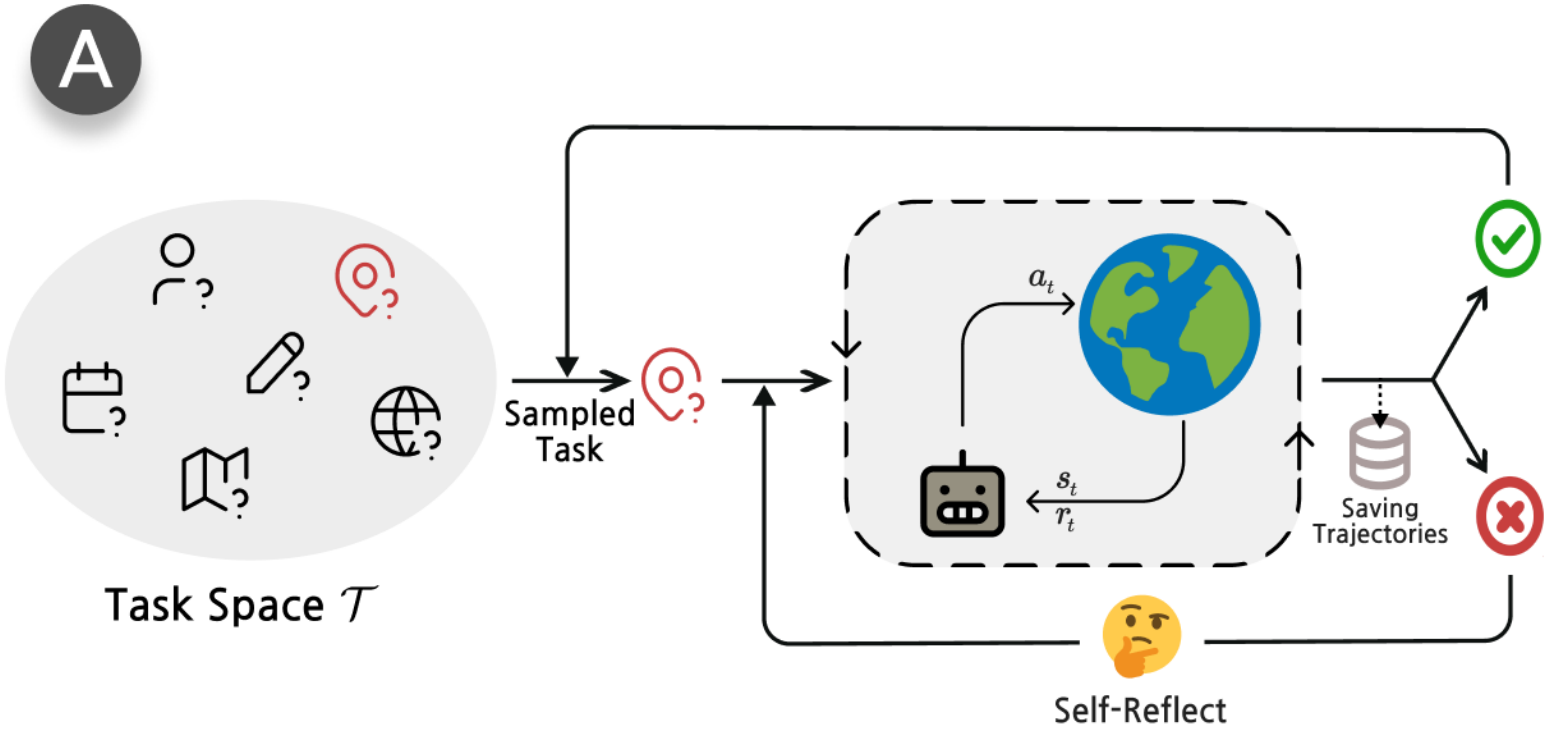
ExpeL Framework





ExpeL Framework

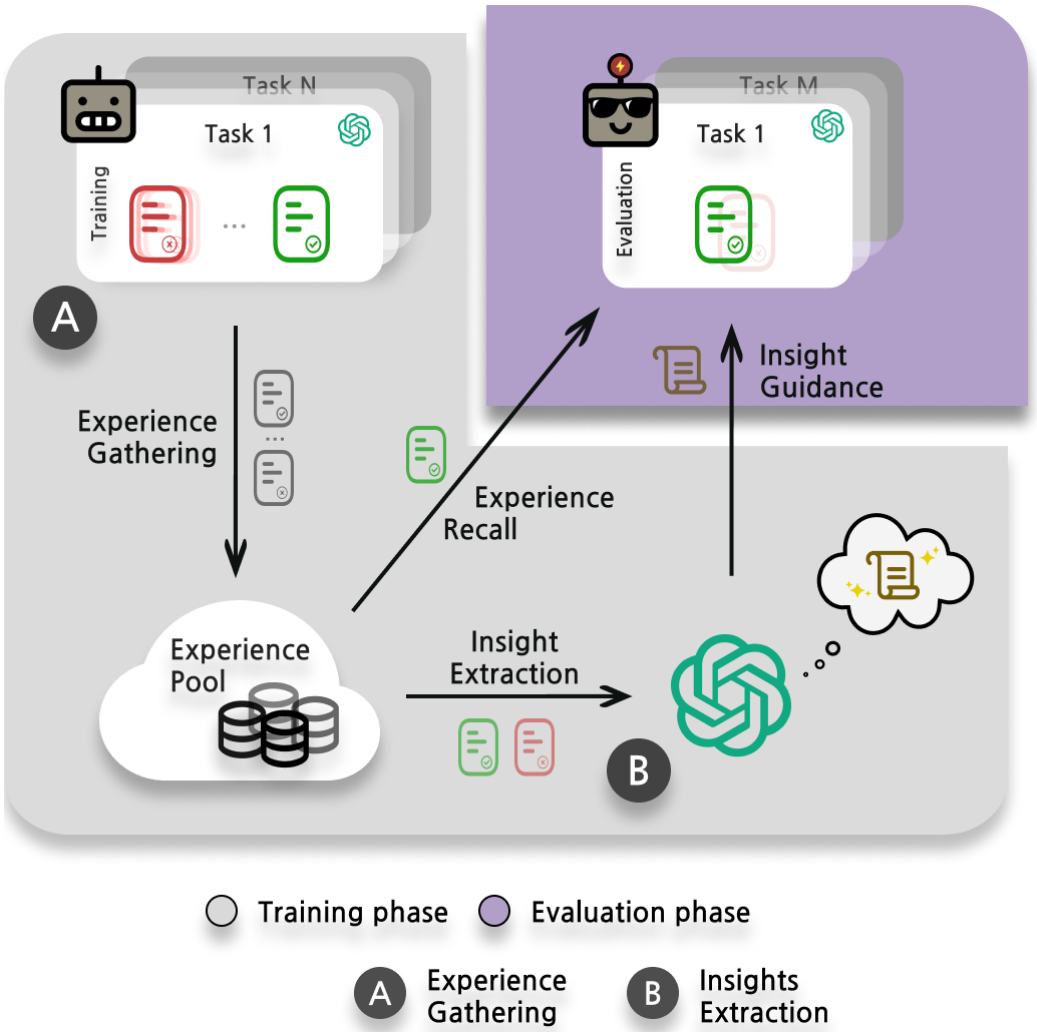
*Gathering Experiences – 정보 추출을 위한 다양한 경험 수집



- LLM은 Few-Shot, Planning, Self-Reflect를 통해 Action을 생성
- 이렇게 축적된 경험은 Insight 추출, Retrieval을 위해 사용

ExpeL Framework

*Learning from Experiences



- 축적한 경험은 Experience Pool에 저장되어 유사한 상황에서 Few-Shot에 추가적으로 사용
- 성공과 실패를 비교하여 올바른 행동과 잘못된 행동 강조
- 성공한 세트에서 패턴을 파악하여 일반적인 성공 관행을 밝힘

ExpeL Framework

*Task Inference

Task description: You are QA system. Solve a question answering task with interleaving ...

Extracted insights:
1. Break down complex queries into simpler, ...
2. Consider that the answer might be in the observations already made...
3. ...

Retrieved in-context examples:

Question:	Which documentary ... ?
Thought 1:	I need to search ...
... :	...
Action N:	Finish[The Saimaa ...]
Obervation N:	Answer is CORRECT

Task: Which episode of SpongeBob ... ?

Task: Which episode of SpongeBob ... ?

Trajectory:

Thought 1:	I need to search ...
Action 1:	Search["The Clash ...]
Observation 1:	"The Clash of Triton", ...
Thought 2:	The paragraph does ...
... :	...
Action H:	Finish["To SquarePants ...]
Obervation H:	Answer is CORRECT

- 추출된 Insight는 항상 동일하게 들어감
- Experience Pool에 저장된 이전 경험들은 RAG 방식을 통해 Task에 유사한 경험을 가지고 옴

ExpeL Framework

*Transfer Learning

Knowledge Finetuning

You are a teacher agent that passes on experience to student agents. You came up with the following rules to help you achieve the task of {Source Task} effectively. The number at the end are the importance you gave to each of the rules.

RULES:

[Extracted insights from Source Task]

Now a student agent is trying to solve a similar {Target Task}.

Some examples of this new task are:

[Fixed fewshot examples of Target Task]

Give a concise and easy to follow instructional paragraph based on the RULES for the student agent to solve {Target Task}. Do not state where each sentence is using whichever rule, and make sure the paragraph is VERY CONCISE and EASY TO FOLLOW!

Knowledge transfer

Fewshot Evaluation

The following paragraph is insights a teacher agent provided to you. It is MANDATORY for you to follow these insights as CLOSELY as possible as they will help you perform the {Target Task} tasks efficiently:

[Finetuned insights]

[Target Task desciption + fewshot]

{Target Task}

- Source Task에서 추출된 Insight는 Target Task의 Few-Shot과 LLM을 통해 Fine-tuning 가능(In-Context Learning)
- 기존에 추출된 Insight만으로도 Target Task를 효율적으로 수행 가능하며, 높은 성공률 달성

Experiments

*Experiments Env

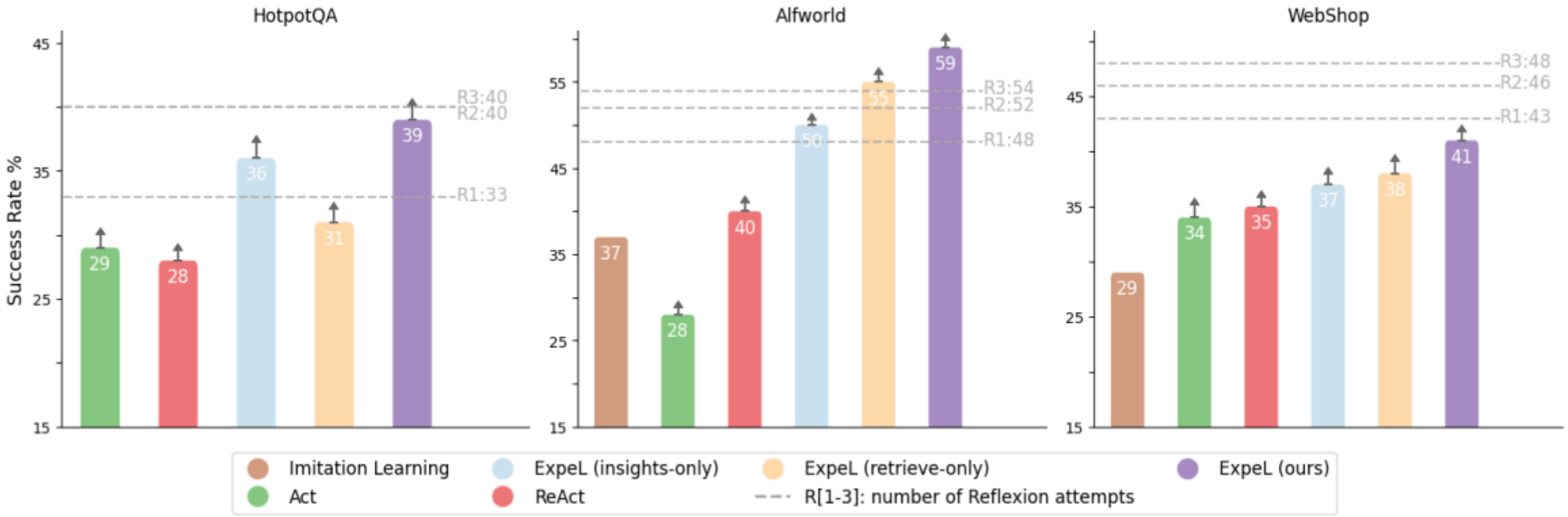
- Hotpot QA
  - 다단계 추론을 통해 여러 문서 탐색, 정답 유추

- Alfworld
  - 언어 기반 시뮬레이션 환경으로 환경 탐색, 목표 달성
  ex) 냄비를 부엌에서 찾아 식탁으로 가져 오세요

- WebShop
  - 온라인 쇼핑 환경에서 상품 검색, 구매 결정
  ex) 별점이 가장 높은 스피커를 하나 찾아 구매하세요

Result

*Experiments Env



Result

*Experiments Env

	HotpotQA (SR %)
ReAct	28.0 ± 1.4
Hand-crafted insights	32.0 ± 1.1
Insights with reflections	29.0 ± 0.4
`gpt-3.5-turbo` insights	32.0 ± 0.4
ExpeL (ours)	**39.0** ± 1.7
	ALFWorld (SR %)
ReAct	40.0 ± 0.3
Reasoning similarity	48.5 ± 2.1
Random sampled	42.5 ± 0.8
ExpeL (ours)	**59.0** ± 0.3

Table 3: **Ablations Results.** *Upper:* **Ablations on insight extraction.** Hand-crafted insights enjoyed a performance boost over ReAct but were less effective than LLM-generated ones. Furthermore, adding reflections to the insight-generating process hurt performance. Lastly, better LLM base models give better insights. *Lower:* **Ablations on in-context examples selection strategy.** Randomly selected baseline has a significant drop in performance while ranking using reason similarity also has a noticeable dip.

Conclusion and Limitations

- 다양한 실험의 부재, 복잡한 환경에서 추가 검증 필요
- Open AI API 의존성

- 경험 학습과 데이터 재사용을 통해 LLM Agent가 학습 가능하다는 것을 보여줌

AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation

Qingyun Wu[†], **Gagan Bansal**[*], **Jieyu Zhang**[±], **Yiran Wu**[†], **Beibin Li**[*]

Erkang Zhu[*], **Li Jiang**[*], **Xiaoyun Zhang**[*], **Shaokun Zhang**[†], **Jiale Liu**[∓]

Ahmed Awadallah[*], **Ryen W. White**[*], **Doug Burger**[*], **Chi Wang**[*1]

[*]Microsoft Research, [†]Pennsylvania State University

[±]University of Washington, [∓]Xidian University

ICLR 2024

Natural Language Processing & Artificial Intelligence

KOREA UNIVERSITY 고려대학교 KOREA UNIVERSITY

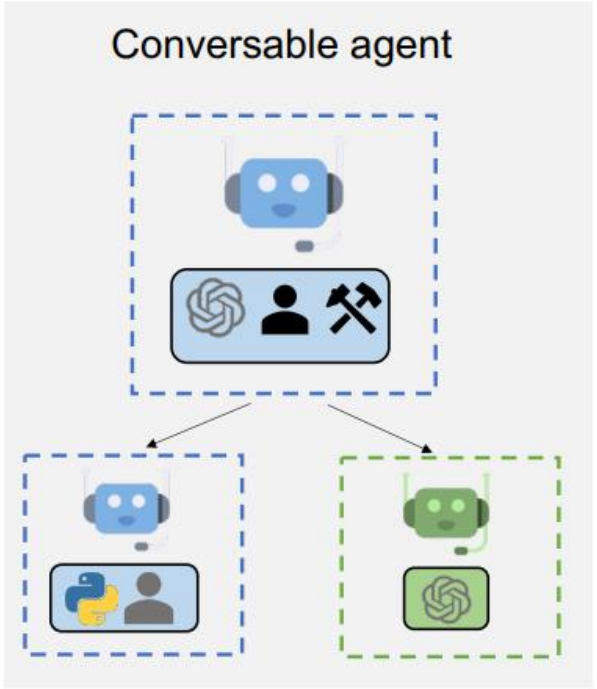
Abstract & Introduction

Task의 범위가 확장되고, 복잡성이 증가함에 따라 Agent 성능 향상 필요

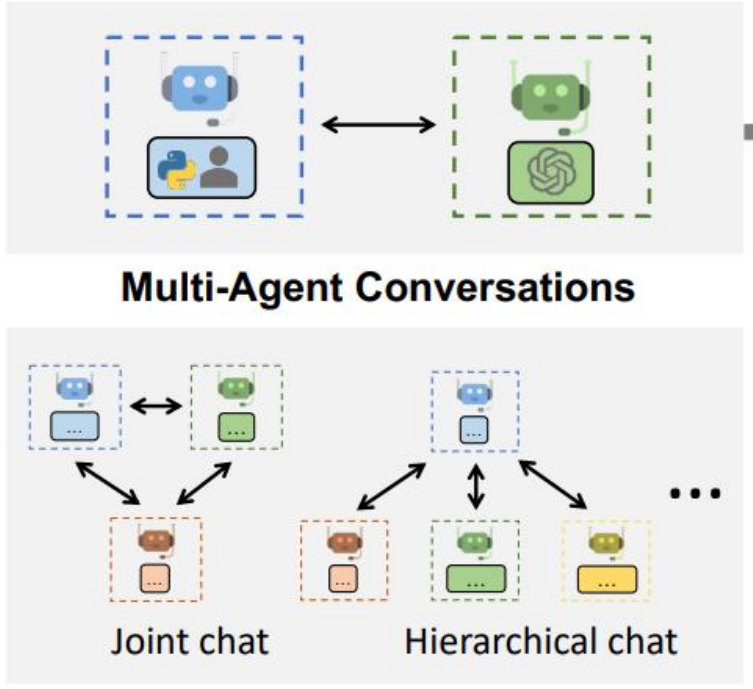
1. LLM은 피드백을 수용, 대화를 통해 협력 가능
2. 올바른 프롬프트를 사용했을 때 다양한 Task 가능
3. 복잡한 작업을 단순한 하위 작업으로 나누었을 때 문제 해결 능력 증가

=> Multi-Agent System

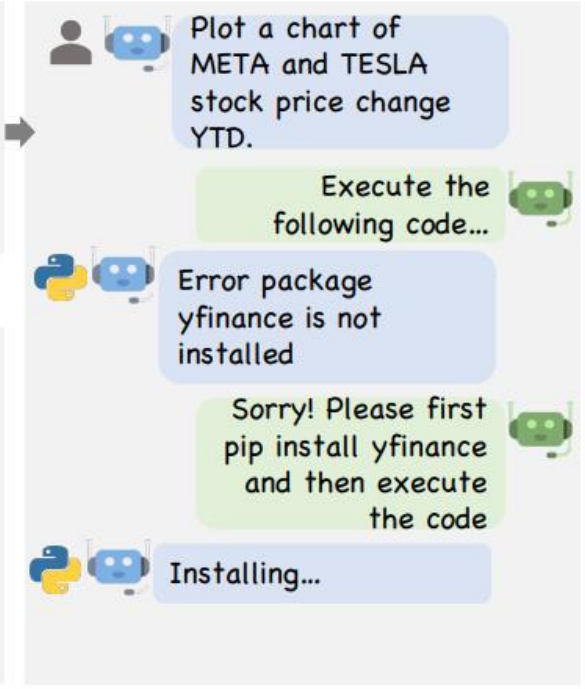
AutoGen Framework



Agent Customization



Flexible Conversation Patterns

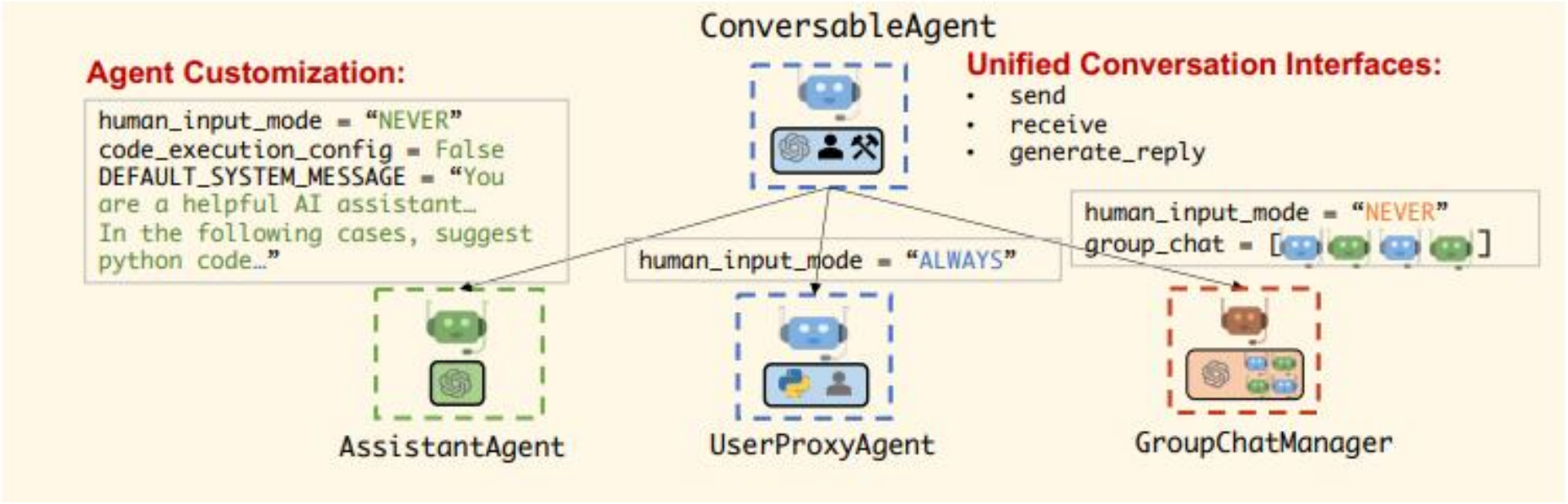


Example Agent Chat

AutoGen Framework

*Conversable Agent - Admin

- 대화 혹은 Task를 진행하면서 Agent, Tool을 호출
- 사전에 정해진 규칙, 패턴에 따라 문제 해결을 진행

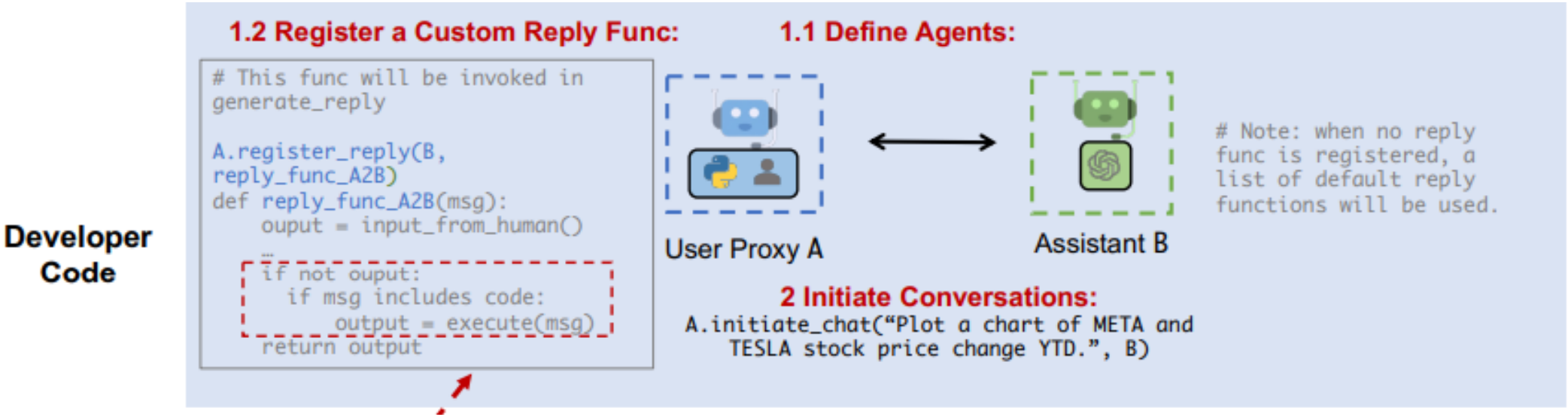


복잡한 문제를 해결하려면 규칙 기반에서 벗어나 Multi-Agent Conversation을 진행할 수 있어야 함

AutoGen Framework

*Conversation Programming – Controlled by LLM, Programming

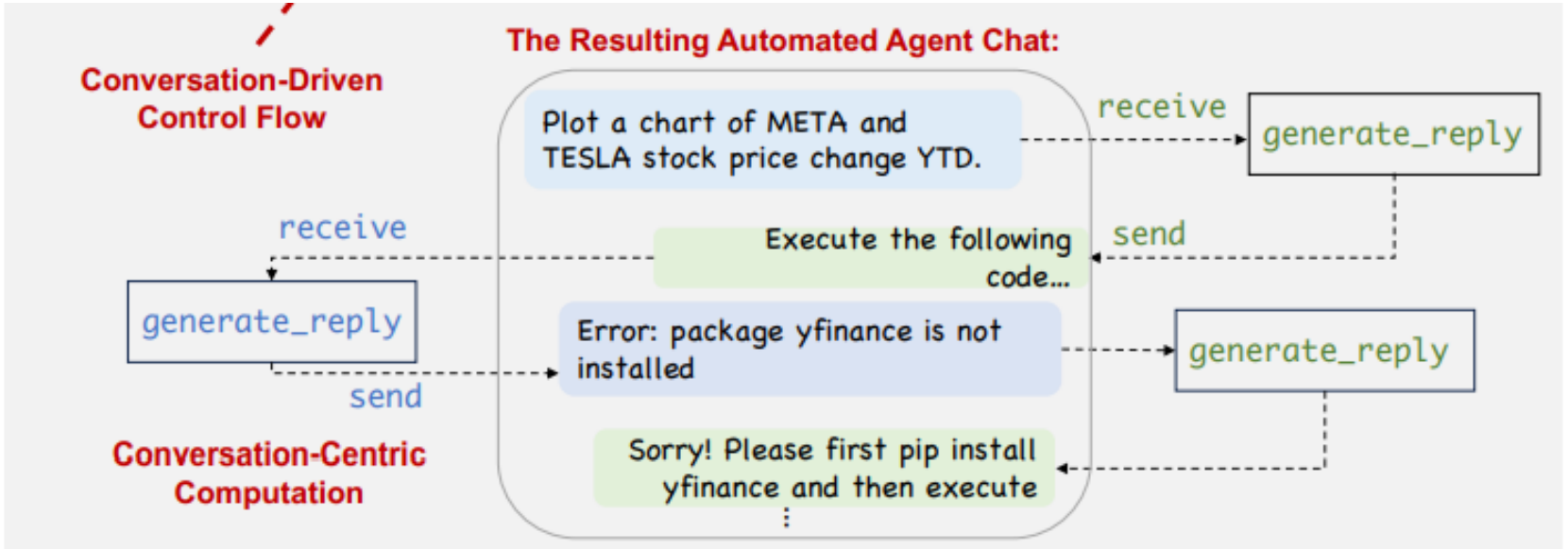
- 자연어 기반, 혹은 프로그래밍으로 동적인 대화를 진행
- Code에서 오류가 발생하더라도 능동적으로 대화를 진행하여 수정 가능



AutoGen Framework

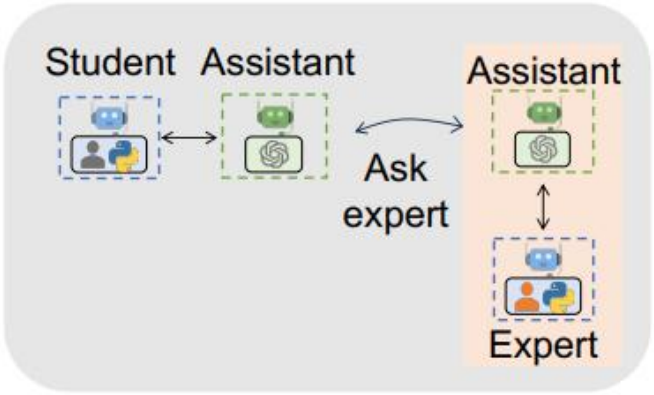
*Conversation Programming – Conversation-centric computation

- 대화를 중심으로 Agent끼리 자동 응답 진행하며, 종료 조건이 만족 될 때 까지 진행
- 사용자가 대화에 끼어들 수 있고, 제어 모듈 없이 자율적인 진행 가능

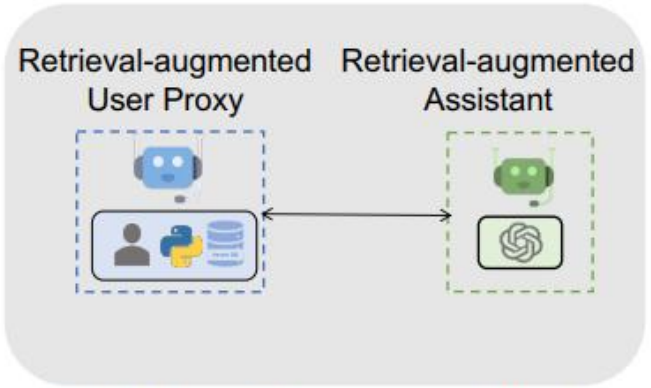


Experiments

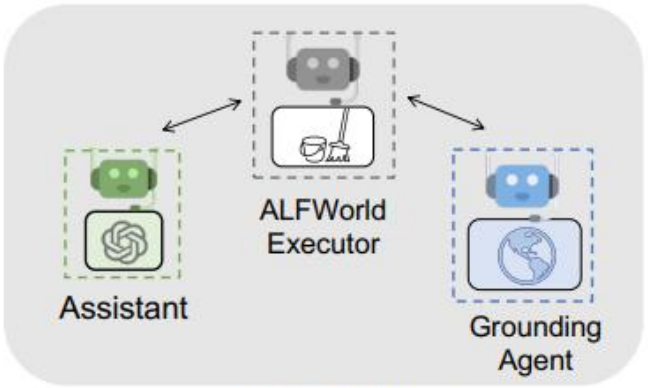
*Examples



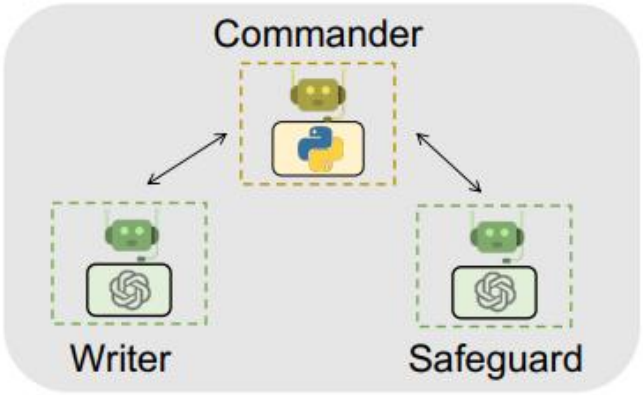
A1. Math Problem Solving



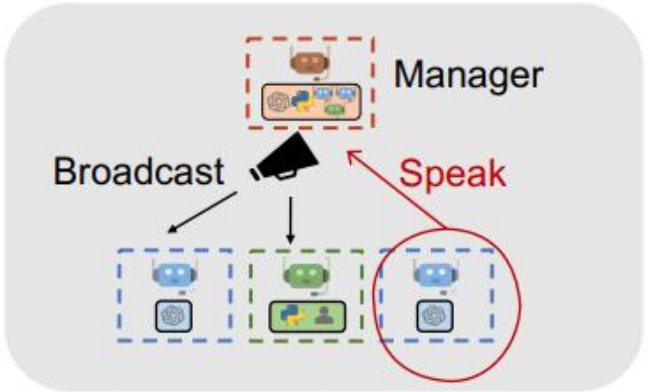
A2. Retrieval-augmented Chat



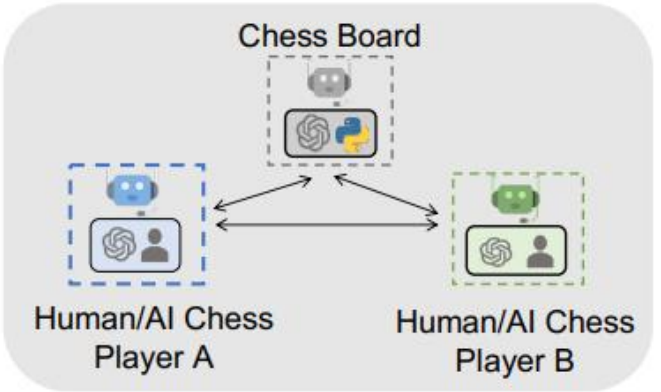
A3. ALF Chat



A4. Multi-agent Coding



A5. Dynamic Group Chat



A6. Conversational Chess

Experiments

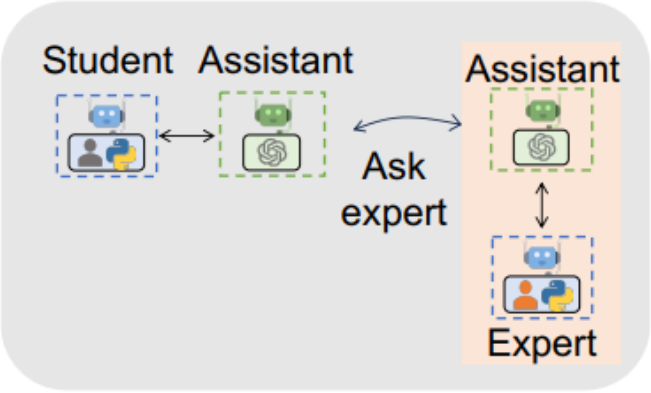
*Examples – Math Problem Solving

	Correctness	Failure Reason
AutoGen	3/3	N/A.
AutoGPT	0/3	The LLM gives code without the print function so the result is not printed.
ChatGPT+Plugin	1/3	The return from Wolfram Alpha contains 2 simplified results, including the correct answer, but GPT-4 always chooses the wrong answer.
ChatGPT+Code Interpreter	2/3	Returns a wrong decimal result.
LangChain ReAct	0/3	LangChain gives 3 different wrong answers.
Multi-Agent Debate	0/3	It gives 3 different wrong answers due to calculation errors.

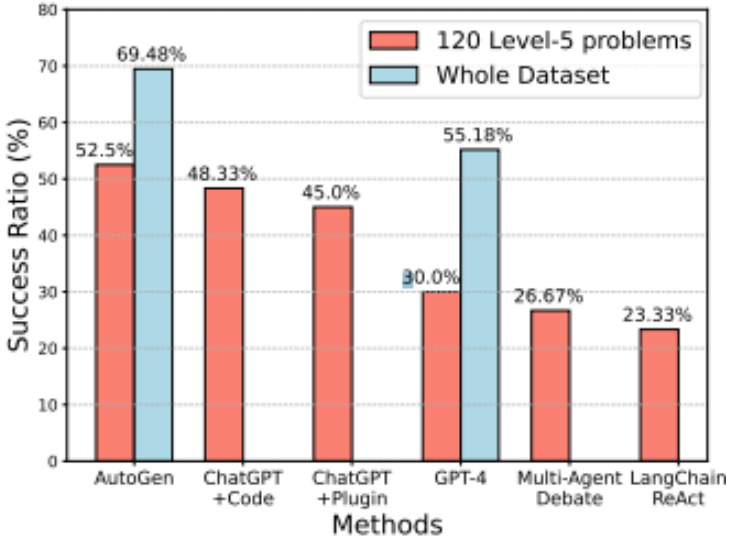
(a) Evaluation on the first problem that asks to simplify a square root fraction.

	Correctness	Failure Reason
AutoGen	2/3	The final answer from code execution is wrong.
AutoGPT	0/3	The LLM gives code without the print function so the result is not printed.
ChatGPT+Plugin	1/3	For one trial, GPT-4 got stuck because it keeps giving wrong queries and has to be stopped. Another trial simply gives a wrong answer.
ChatGPT+Code Interpreter	0/3	It gives 3 different wrong answers.
LangChain ReAct	0/3	LangChain gives 3 different wrong answers.
Multi-Agent Debate	0/3	It gives 3 different wrong answers.

(b) Evaluation on the second number theory problem.

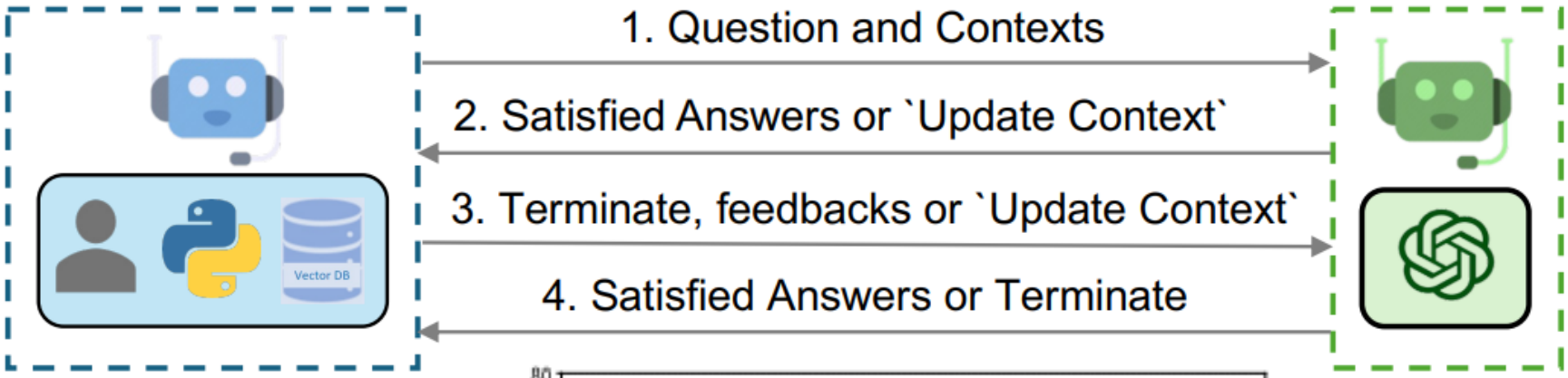


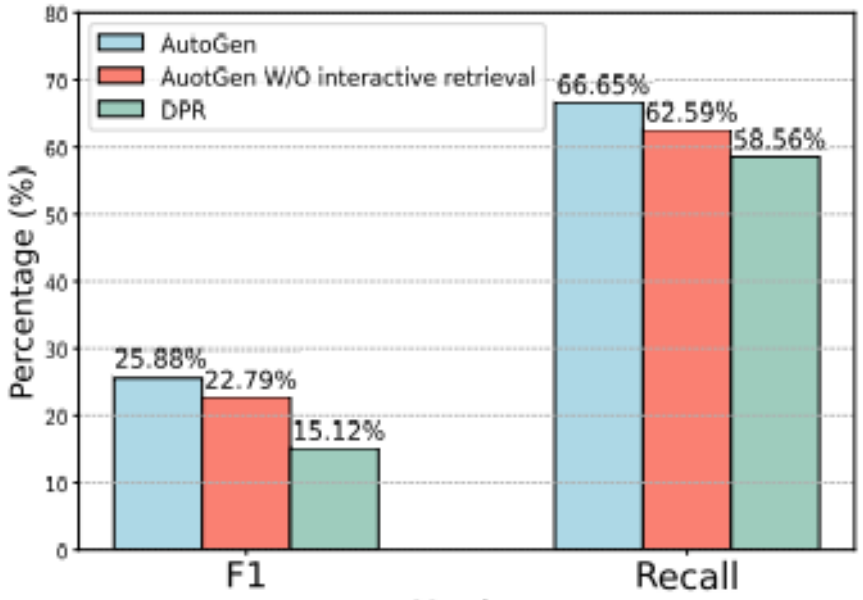
A1. Math Problem Solving



Experiments

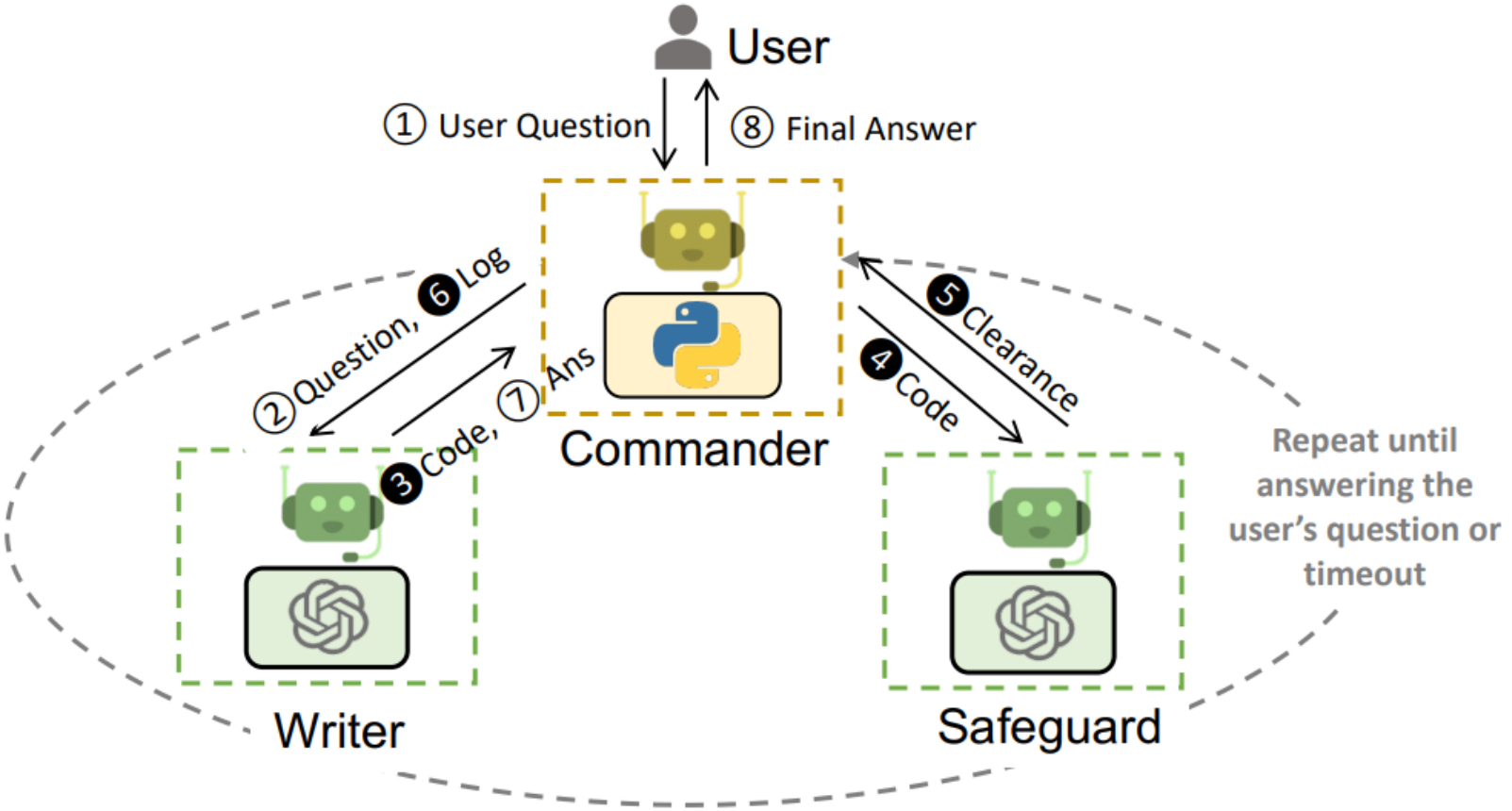
*Examples – Retrieval Augmented Chat

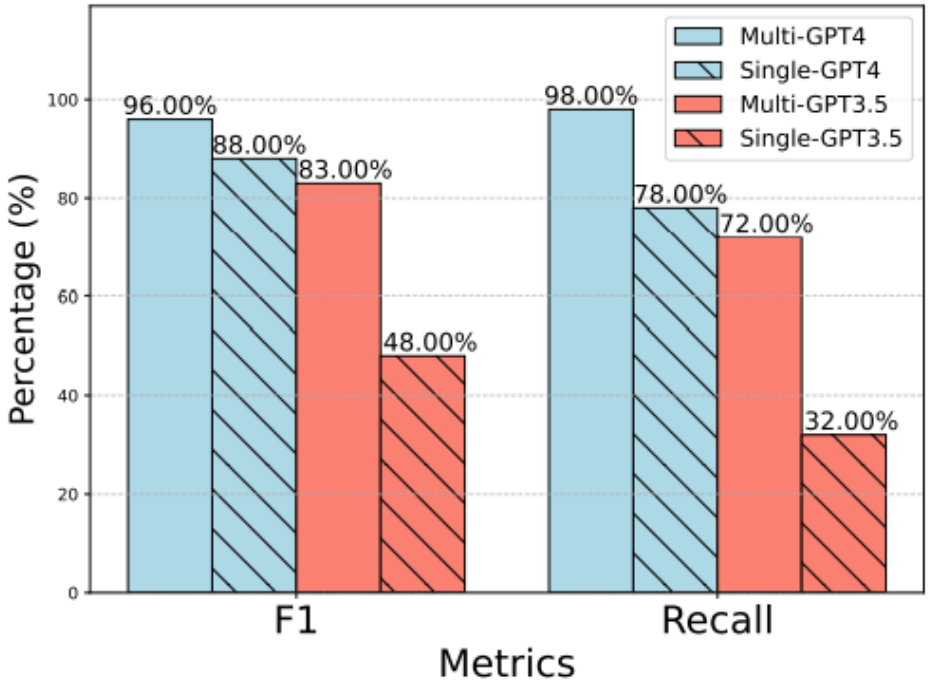




Experiments

*Examples – Multi Agent Coding





Dataset	netflow	facility	tsp	coffee	diet
Saving Ratio	3.14x (0.65)	3.14x (0.64)	4.88x (1.71)	3.38x (0.86)	3.03x (0.31)

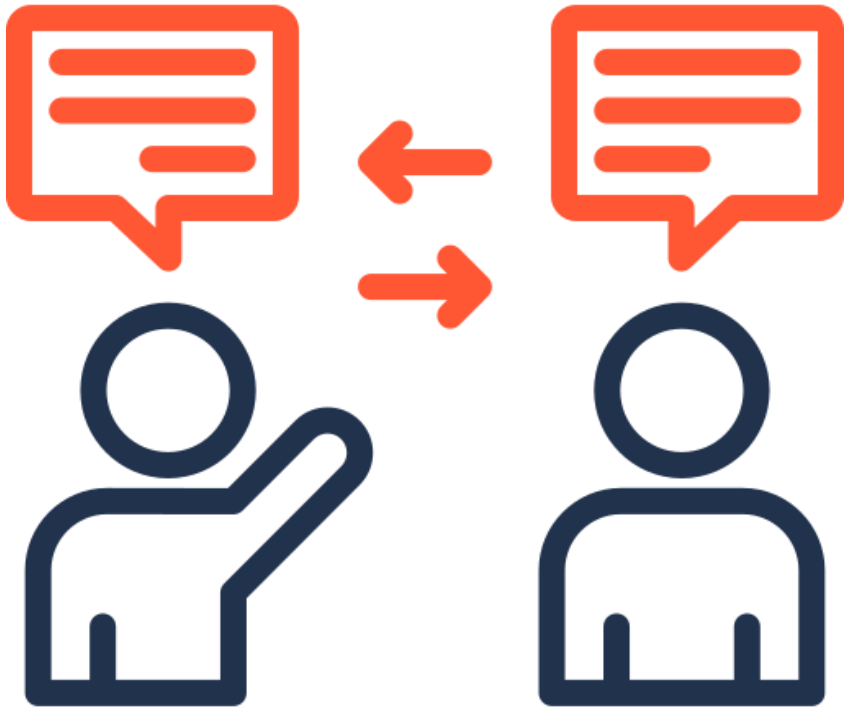
Conclusion and Limitations

- 대화가 연속적으로 길어지면서 Long-Context 문제가 발생
- 다양한 모델을 활용하여 실험을 진행하지 않음

- Multi Agent System을 통해 생산성, 효율성을 증가

Negotiation AI Agent

*기존 협상 과정의 문제점

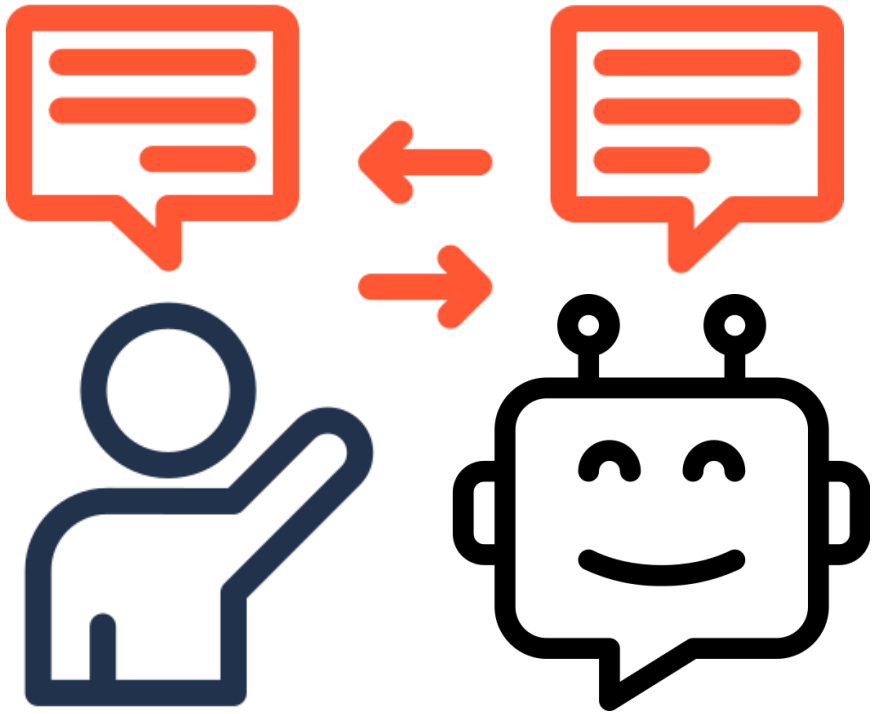


- 시간 소모가 큼
  -> 복잡한 조건 협상으로 인해 장기간 협상 지속
- 객관성 부족
  -> 감정적 판단이 개입되어 최적의 결과 도출 어려움
- 복잡한 조건 처리 문제
  -> 여러 이해관계자 간의 조건 조율 어려움
- 다양한 데이터 활용 저조
  -> 과거 협상 데이터를 활용하지 못해 비효율 발생

=> LLM을 활용한 AI Agent 활용으로 최적의 협상 진행

Negotiation AI Agent

*Negotiation AI Agent의 중요성 및 활용 가능성



- 효율성 증가
  -> 복잡한 협상 자동화로 시간과 자원 절약
- 객관성 제공
  -> 감정적 판단에서 벗어나 객관적이고 공정한 결과 도출
- 복잡한 조건 처리 최적화
  -> 정보의 비대칭성을 활용한 최고의 이득 제안
- 데이터 기반 결정
  -> 과거 데이터를 통해 최적의 전략 제안

=> AI Agent는 협상을 효율화하고 공정성을 높임

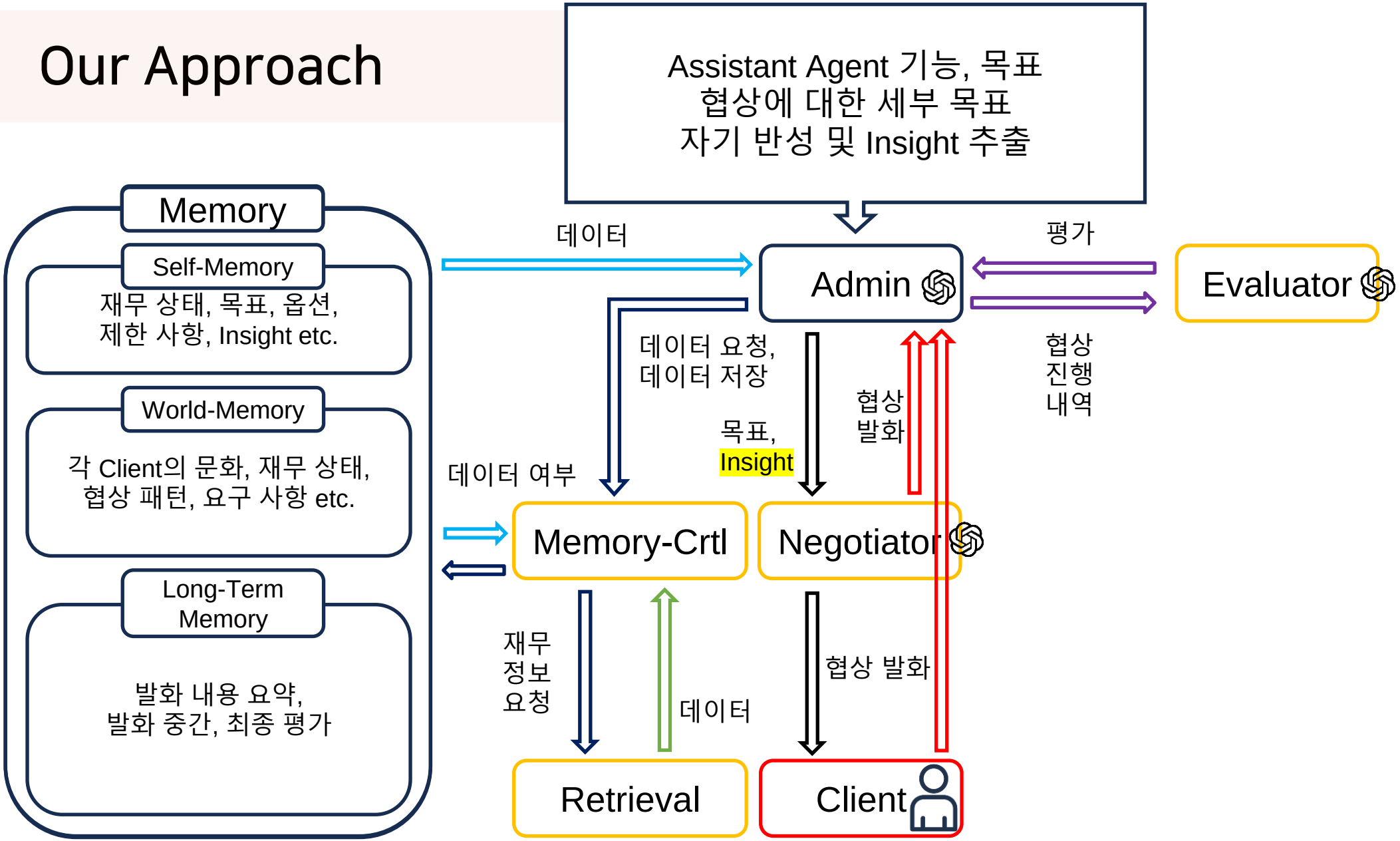
Our Approach

***Negotiation AI Agent의 달성 목표**

- 기존 단순한 협상 진행에서 다양한 데이터를 활용한 동적인 협상 진행으로 변화
- Critic의 역할을 극대화 하여 LLM의 발화 능력을 최대한 활용
- 상대방의 데이터를 저장하여 다음 협상에는 유리한 방향으로 협상 진행
- Insight 추출을 통해 짧은 Context로도 발화 능력 강화
- Open Source인 KULLM과 초 거대 언어 모델인 GPT를 섞어서 사용하여 API 요금 감소, GPU 과부하 감소

=> **Multi Agent System**을 통해 Negotiation의 성공률, 이득을 극대화

Our Approach



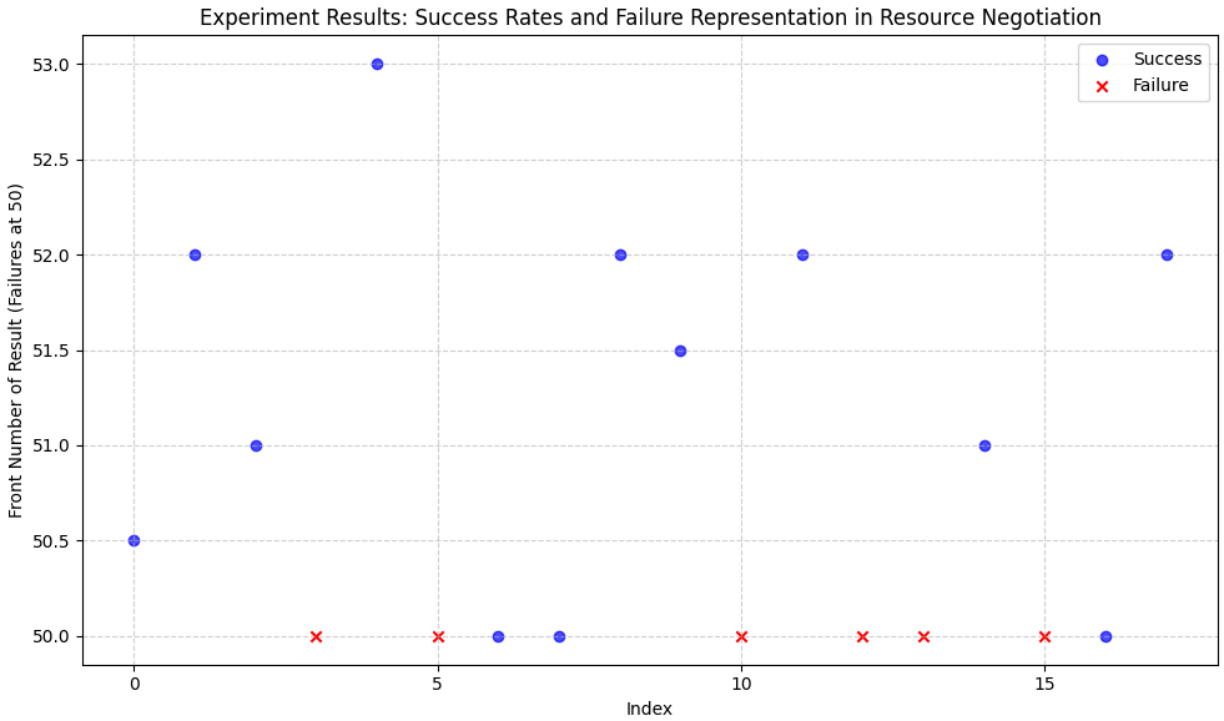
Experiment

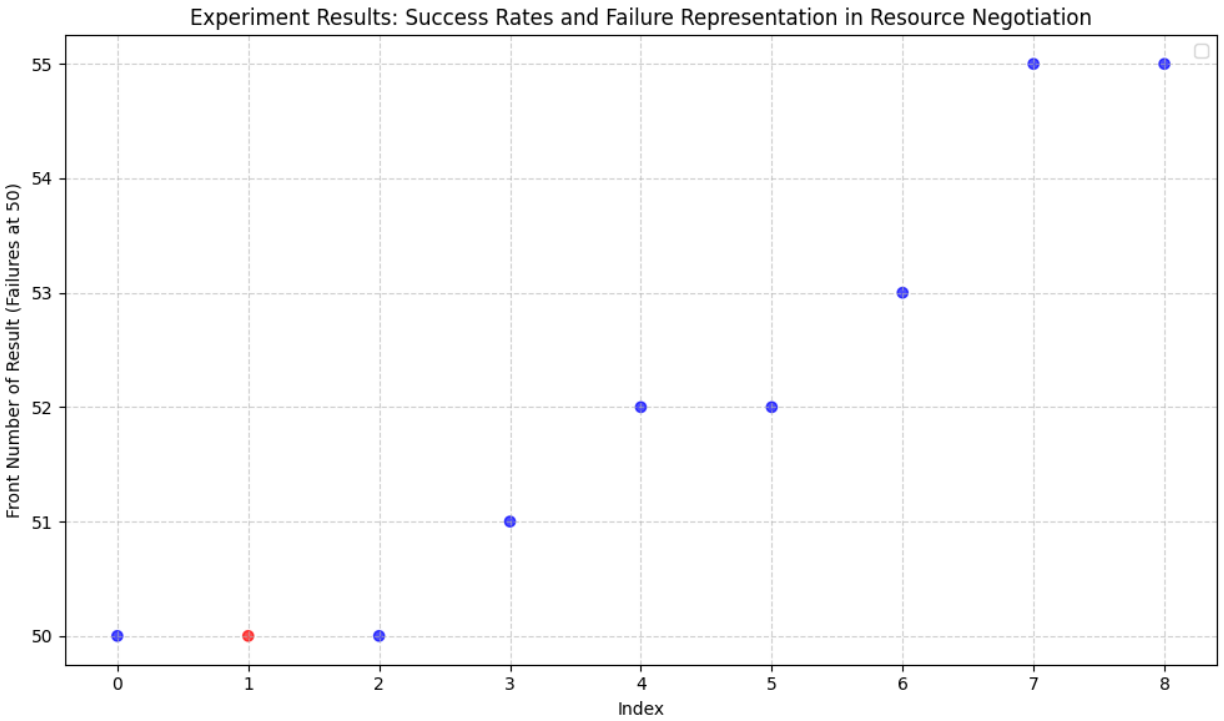
*Negotiation AI Agent의 성능 평가

- 100이라는 공동 자원을 두고, Red팀과 Blue 팀이 나눠 가지는 환경
- 협상은 최대 5턴이 진행되고, Red에 Negotiation AI Agent가 활용
- 추후 정해진 가격을 두고 얼마나 비싸게 팔 수 있는지, 싸게 살 수 있는지 실험 진행

Experiment

*Negotiation AI Agent의 성능 평가





Thank you
Q&A