LLM Agent 개관

문현석

어떻게 정의되고, 어떻게 만들고, 어떻게 평가하고

LLM Agent

ICLR2024

AGENTBENCH: EVALUATING LLMS AS AGENTS

Xiao Liu[1,*], Hao Yu[1,*,†], Hanchen Zhang[1,*], Yifan Xu[1], Xuanyu Lei[1], Hanyu Lai[1], Yu Gu[2,†], Hangliang Ding[1], Kaiwen Men[1], Kejuan Yang[1], Shudan Zhang[1], Xiang Deng[2], Aohan Zeng[1], Zhengxiao Du[1], Chenhui Zhang[1], Sheng Shen[3], Tianjun Zhang[3], Yu Su[2], Huan Sun[2], Minlie Huang[1], Yuxiao Dong[1,‡], Jie Tang[1,‡]

[1]Tsinghua University, [2]The Ohio State University, [3]UC Berkeley

https://openreview.net/pdf?id=zAdUB0aCTQ

+ 기타등등

NeurIPS2023

Language Models can Solve Computer Tasks

Geunwoo Kim
University of California, Irvine
kgw@uci.edu

Pierre Baldi
University of California, Irvine
pfbaldi@ics.uci.edu

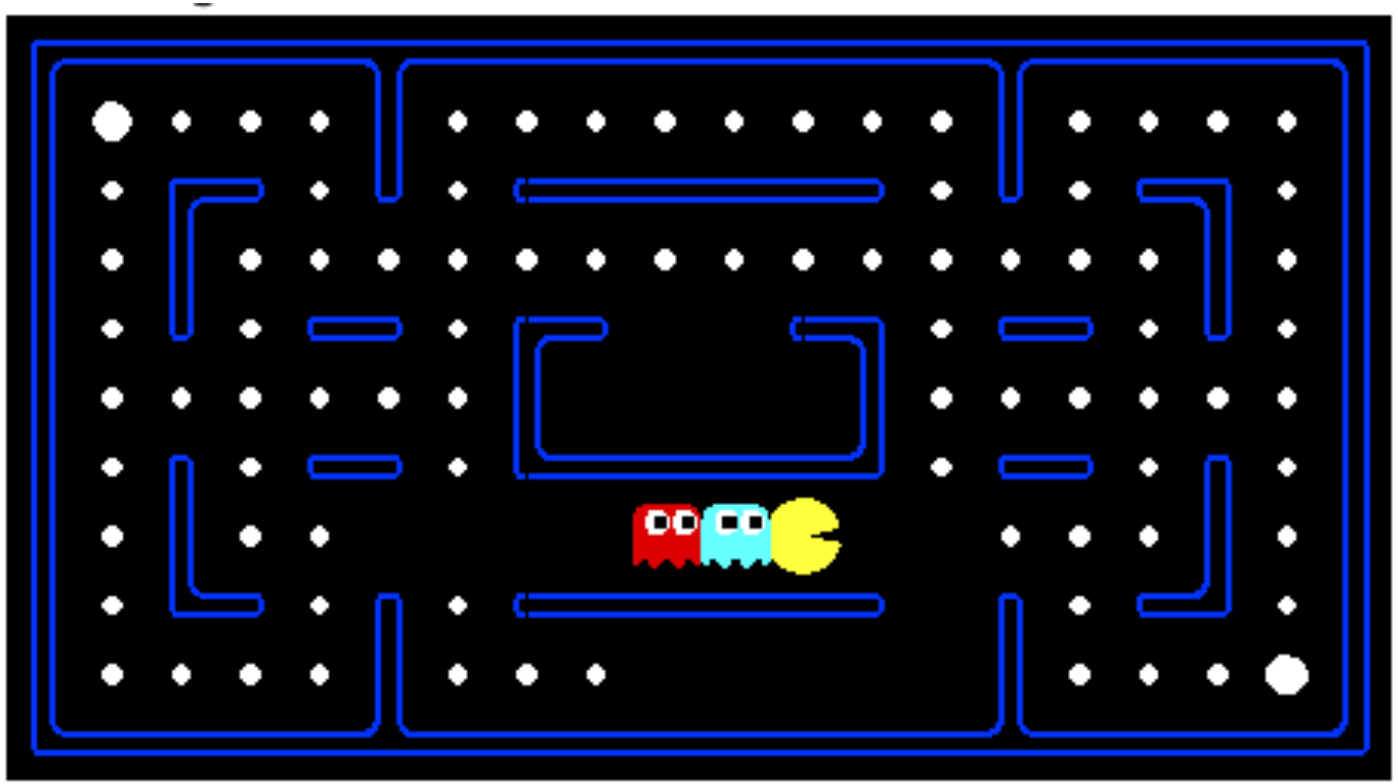
Stephen McAleer*
Carnegie Mellon University
smcaleer@cs.cmu.edu

https://proceedings.neurips.cc/paper_files/paper/2023/file/7cc1005ec73cfbaac9fa21192b622507-Paper-Conference.pdf

Simplified Example

AI Agent는 뭔가

Goal: **팩맨**을 움직여서 유령을 피하고 최고 점수를 얻자



Simplified Example

AI Agent는 뭔가

Agent: 팩맨

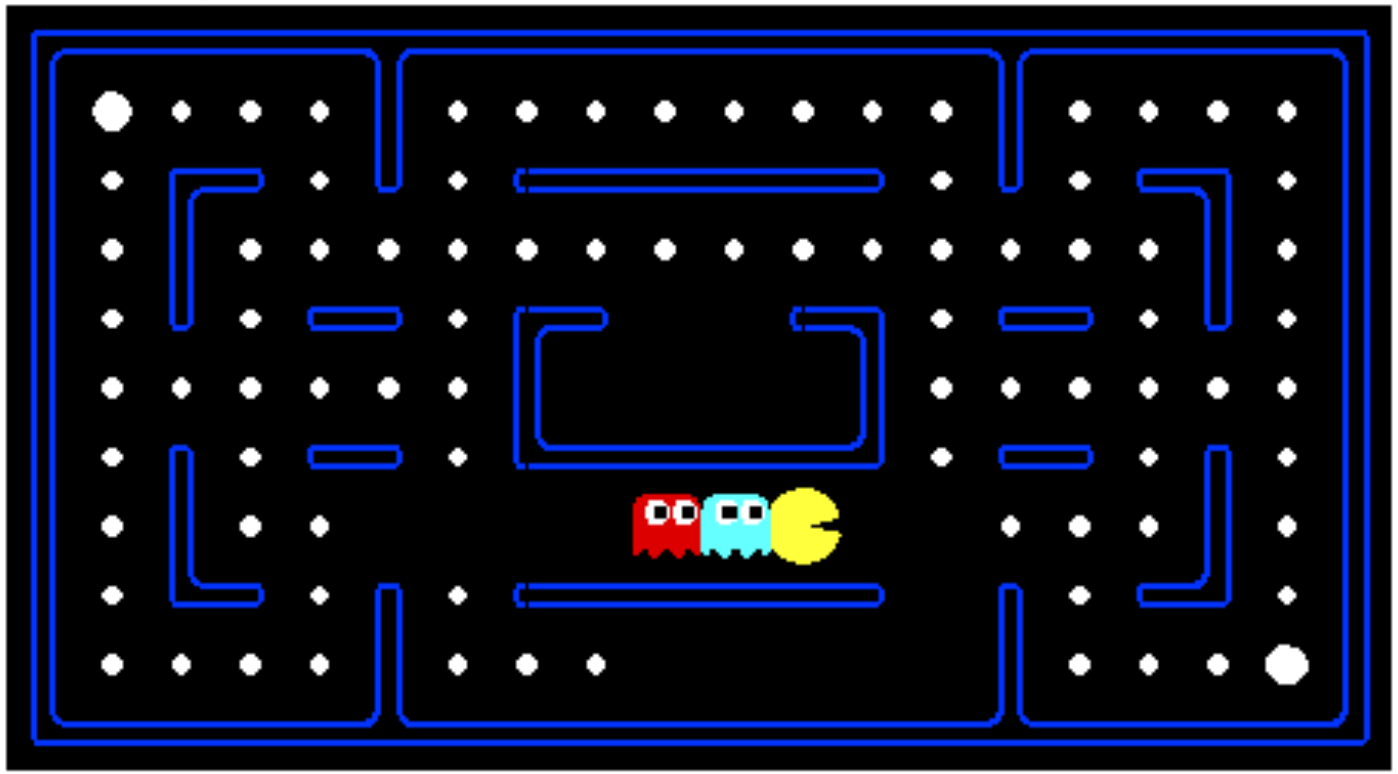
State Space: 점수는 어디있고 유령은 어디있는지

Action Space: 상/하/좌/우

State Space 정보를 참고하여

Agent의 행동을 Action space중에서 고르면서

최고점수를 얻자



수식으로 정의하기

LLM Agent는 뭔가

LLM-as-Agent could be regarded as a **Partially Observable Markov Decision Process**

$$(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{U}, \mathcal{O})$$

- $\mathcal{S}$: State Space
- $\mathcal{A}$: Action Space
- $\mathcal{T}$: Transition Function $\mathcal{T}: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$
- $\mathcal{R}$: Reward assigning function
- $\mathcal{U}$: Task Instruction Space
- $\mathcal{O}$: Observation Space

수식으로 정의하기

LLM Agent는 뭔가

LLM-as-Agent could be regarded as a **Partially Observable Markov Decision Process**

$$(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{U}, \mathcal{O})$$

- $\mathcal{S}$: State Space — 팩맨이 움직일 수 있는 공간
- $\mathcal{A}$: Action Space — 팩맨이 취할 수 있는 움직임
- $\mathcal{T}$: Transition Function $\mathcal{T}: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ — 팩맨이 움직인 이후 State
- $\mathcal{R}$: Reward assigning function — 점수 먹고 유령 만나면 죽고
- $\mathcal{U}$: Task Instruction Space — Action을 LLM에게 입력하는 방식
- $\mathcal{O}$: Observation Space — LLM이 State Space를 보는 방법

수식으로 정의하기

Real-World Agent

LLM-as-Agent could be regarded as a **Partially Observable Markov Decision Process**

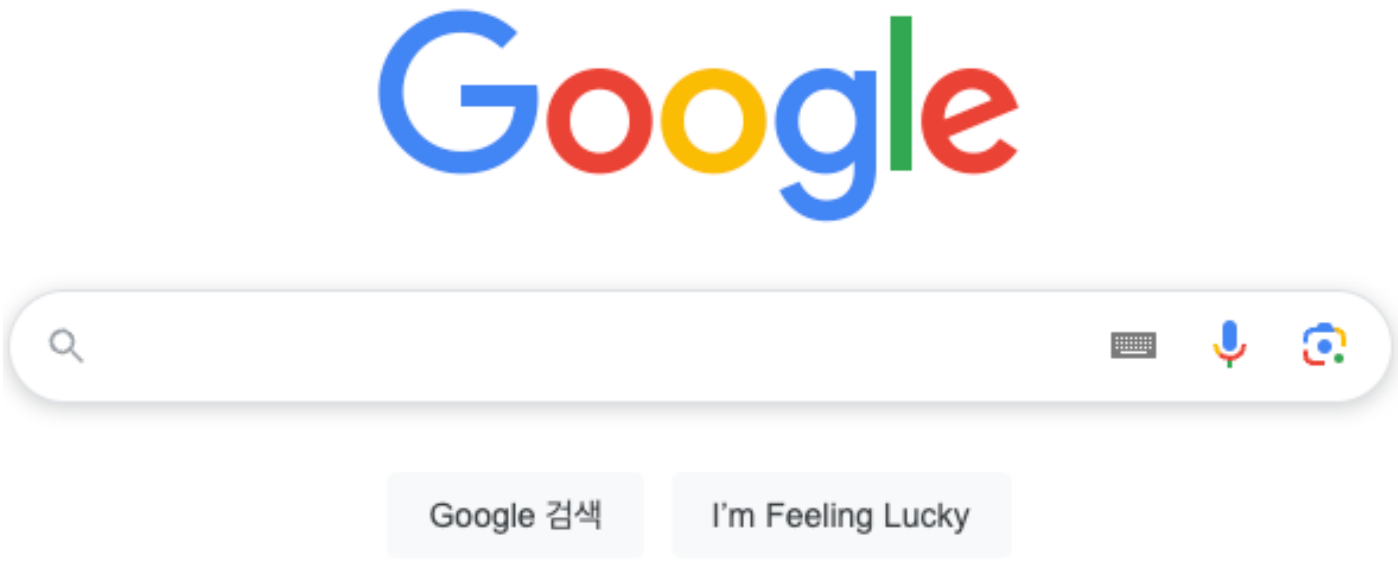
$$(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{U}, \mathcal{O})$$

- $\mathcal{S}$: State Space — Agent에게 주어진 현재 real world상황
- $\mathcal{A}$: Action Space — Agent가 할 수 있는 행동
- $\mathcal{T}$: Transition Function $\mathcal{T}: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ — Agent의 행동 이후 State
- $\mathcal{R}$: Reward assigning function — Agent의 행동에 대한 reward
- $\mathcal{U}$: Task Instruction Space — Action을 LLM에게 입력하는 방식
- $\mathcal{O}$: Observation Space — LLM이 State Space를 보는 방법

수식으로 정의하기

Real-World Agent

예컨대 Web Search Agent라면?



- $\mathcal{S}$: State Space — 지금 바라보는 이 웹 페이지
- $\mathcal{A}$: Action Space — Web Agent로서 할 수 있는 행동들
  (클릭 / 입력 / 엔터키 누르기 등등..)

Text based model은 어떻게 씀?

MiniWoB

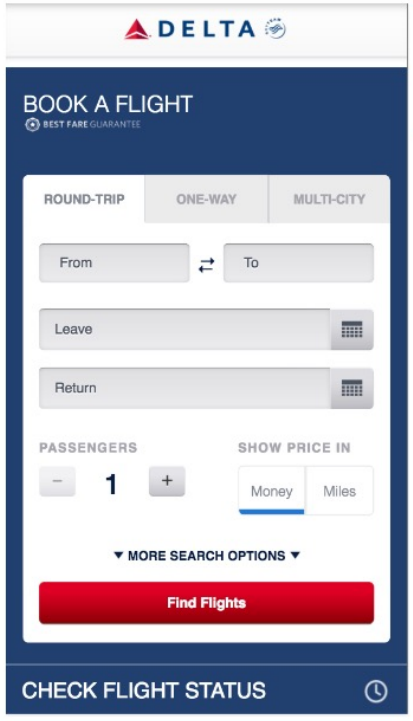
https://arxiv.org/pdf/1802.08802 (ICLR2018)

https://proceedings.mlr.press/v70/shi17a/shi17a.pdf (ICML2017)

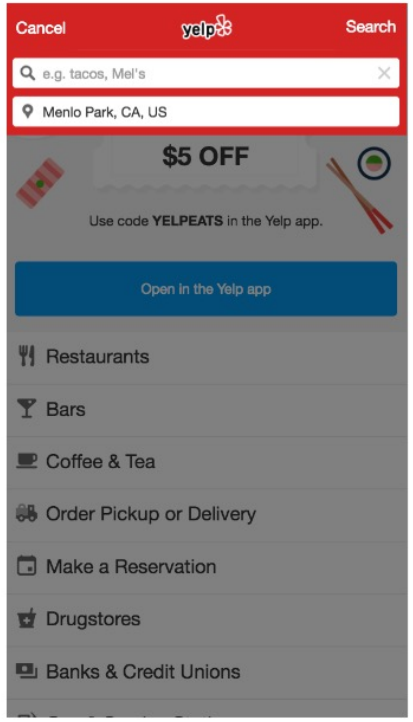
State Space를 html code로 줌

→ Action에 따라 변화하는 html code를 확인

Question: *Can you book a flight from San Francisco to New York?*



Question: *What is the top rated place to eat Korean food in SF?*



명확한 홈페이지 + HTML 코드

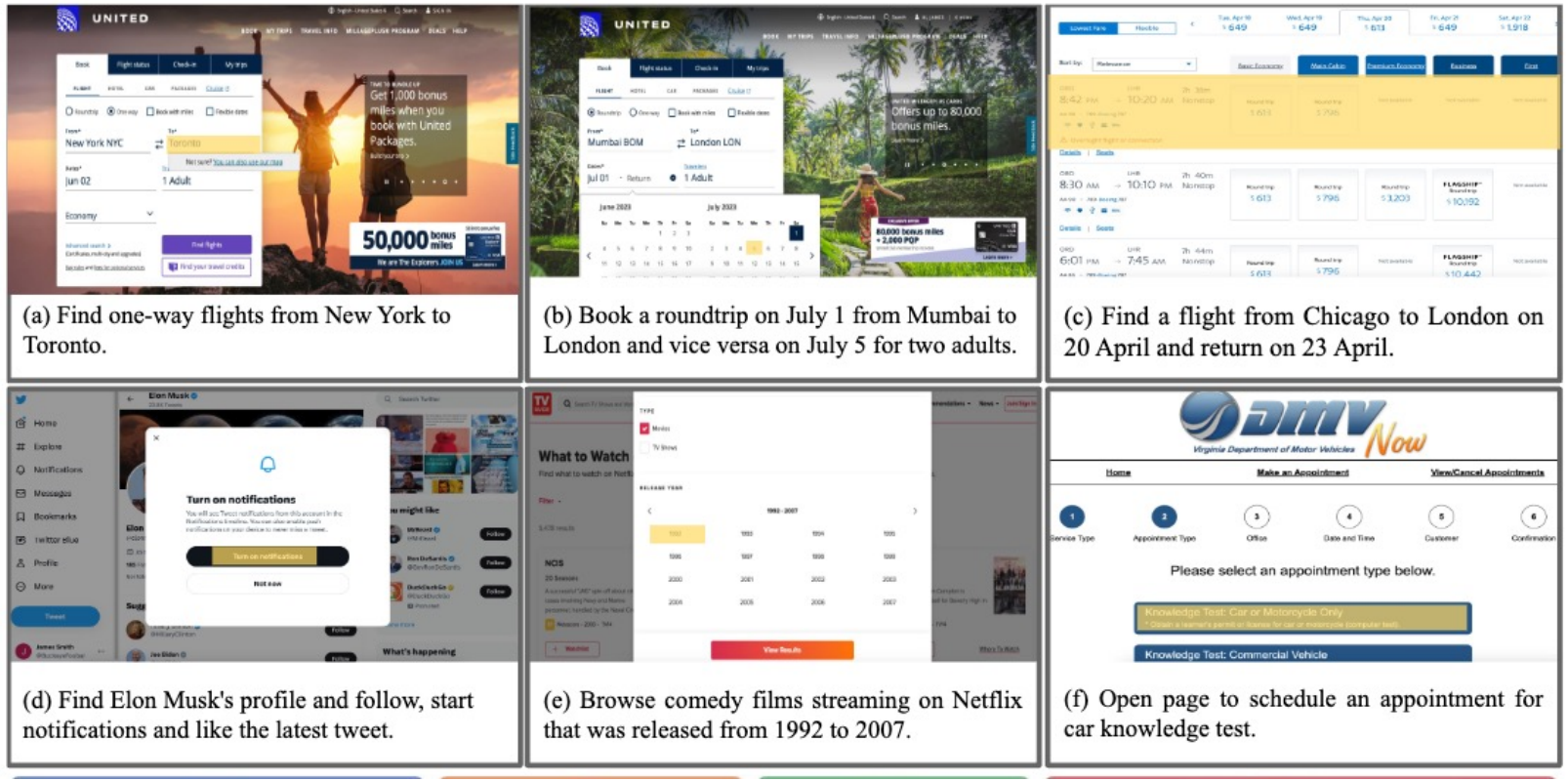
해당 페이지 내에서 할 수 있는 명확한 Question


```
'{action: mouseup, ref: 4}{action: mouseup, ref: 4}{action:
mousedown, ref: 1}{action: mousedown, ref: 1} Expand the
section below and click submit. <BODY ref="1"><DIV id="wrap"
ref="2"><DIV id="area" classes="ui-accordion ui-widget ui-
helper-reset" ref="3"><H3 id="ui-id-1" classes="ui-accordion-
header ui-corner-top ui-accordion-header-collapsed ui-corner-
all ui-state-default ui-accordion-icons ui-state-hover ui-
state-focus" ref="4" recordingTarget="True"><t ref="-5"
text="Section #29"></t></H3><P id="ui-id-3" classes="ui-
accordion-header ui-corner-top ui-accordion-header-collapsed
ui-corner-all ui-state-default ui-accordion-icons" ref="5">
<BUTTON id="subbtn" classes="secondary-action" ref="6"
text="Submit"></BUTTON></P></DIV></DIV></BODY>'
```

Text based model은 어떻게 씀?

MIND2WEB

https://proceedings.neurips.cc/paper_files/paper/2023/file/5950bf290a1570ea401bf98882128160-Paper-Datasets_and_Benchmarks.pdf#page=0.97 (NIPS 2023)



(a) Find one-way flights from New York to Toronto.

(b) Book a roundtrip on July 1 from Mumbai to London and vice versa on July 5 for two adults.

(c) Find a flight from Chicago to London on 20 April and return on 23 April.

(d) Find Elon Musk's profile and follow, start notifications and like the latest tweet.

(e) Browse comedy films streaming on Netflix that was released from 1992 to 2007.

(f) Open page to schedule an appointment for car knowledge test.

일반적으로,

State Space를 HTML로 구성

MiniWoB:

Type New York in the location field, click the search button and choose the tomorrow tab,

MIND2WEB:

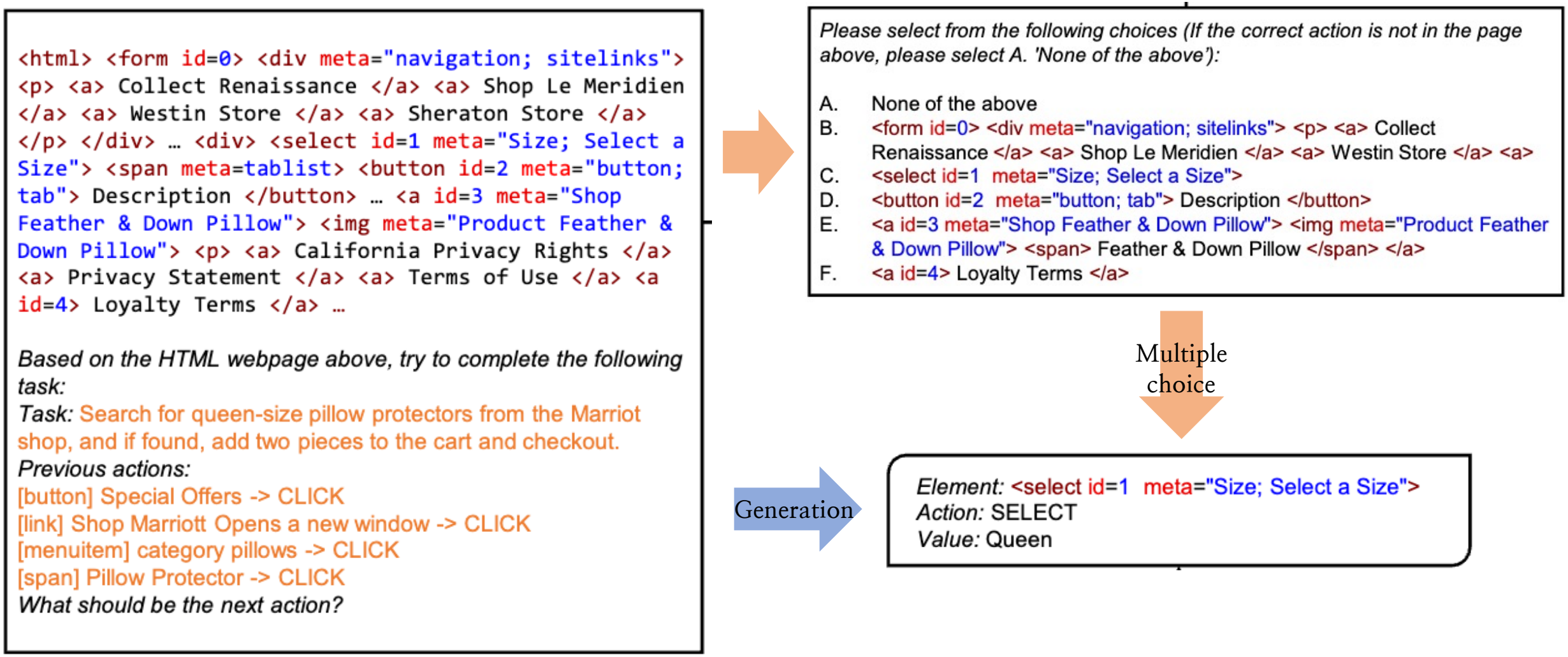
What is the weather for New York tomorrow?

Text based model은 어떻게 씀?

MIND2WEB

https://proceedings.neurips.cc/paper_files/paper/2023/file/5950bf290a1570ea401bf98882128160-Paper-Datasets_and_Benchmarks.pdf#page=0.97 (NIPS 2023)

LLM Agent를 활용하는 방법 (예시)



Text based model은 어떻게 씀?

MIND2WEB

https://proceedings.neurips.cc/paper_files/paper/2023/file/5950bf290a1570ea401bf98882128160-Paper-Datasets_and_Benchmarks.pdf#page=0.97 (NIPS 2023)

LLM Agent를 활용하는 방법 (예시)

We have an autonomous computer control agent that can perform a set of instructions to control computers. First, given the instruction that matches the regular expression, <type regex>, it can type a list of characters via the keyboard. This instruction should specify the target keyboard input for the agent to type. Before this typing instruction, you should first locate the cursor by clicking the input box with the click instruction. Second, given the instruction that matches the regular expression, <press regex>, it can press a specific key on the keyboard. Third, given the instruction that matches the regular expression, <clickoption regex>, it can click an option HTML element in a list with an XPath that is visible on the webpage. The target of this instruction should be a valid XPath. Fourth, given the instruction that matches the regular expression, <movemouse regex>, it can move the mouse cursor on an HTML element with an XPath that is visible on the webpage. Lastly, given the instruction that matches the regular expression, <clickxpath regex>, it can click an HTML element with an XPath that is visible on the webpage. The target of this instruction should be a valid XPath.

Table 6: Agent specification.

```
<type regex> = "^type\s.{1,}$"
<press regex> = "^press\s(enter|arrowleft|arrowright|arrowup|arrowdown|
    backspace)$"
<clickoption regex> = "^clickoption\s.{1,}$"
<movemouse regex> = "^movemouse\s.{1,}$"
<clickxpath regex> = "^clickxpath\s.{1,}$"
```

Listing 1: Regular expressions for specifying the admissible actions.

대전제: Instruction Following Capability

Agent 성능 향상 방안

Recursive Criticism and Improvement

Q: Valerie earns $5000 per month, 1/2 of what her brother earns. If their mother earns twice their combined salary, what's the total amount of money they all have together?

Zero-Shot

A:
Valerie's brother earns $10,000 per month, while Valerie herself earns half of that amount. Together, they earn a total of $15,000 per month. Their mother's monthly earnings amount to $30,000, which is twice the combined salary of Valerie and her brother. Thus, the total amount of money they all have together is $30,000. **(Wrong)**

Few-Shot-CoT

<Few shot examples>
A:
Valerie earns $5000 per month, half of her brother's earnings, making his monthly income $10000. Combined, Valerie and her brother earn $15000 per month. Their mother earns double their combined salary, amounting to $30000 per month. In total, they have $30000 collectively. **(Wrong)**

Zero-Shot-CoT

A: Let's think step by step. First, Valerie's brother earns 2x and Valerie earns x (half of her brother's earnings). Next, their mother's income is twice the combined earnings of Valerie and her brother. In this case, their total income amounts to $30,000. **(Wrong)**

RCI

<Initial output generation>

A: Valerie's brother earns $10,000 per month, while Valerie herself earns half of that amount. Together, they earn a total of $15,000 per month. Their mother's monthly earnings amount to $30,000, which is twice the combined salary of Valerie and her brother. Thus, the total amount of money they all have together is $30,000.

<Critique>

Review your previous answer and find problems with your answer.

All individuals, including Valerie, her brother, and mother, should be taken into account when calculating the overall sum of money.

<Improve>

Based on the problems you found, improve your answer.

Valerie's brother earns $10,000 per month, while Valerie herself earns $5,000. Together, they earn a total of $15,000 per month. Their mother earns double their combined salary, which amounts to $30,000 per month. In total, they have $45,000 collectively. **(Correct)**

Recursive Criticism and Improvement (RCI)

Agent의 입력으로 사용되기 위한 3단계의 reasoning

Task Grounding

: Improves actions to be more effective in solving the given task

State Grounding

: Ensures the feasibility of actions in the current state

Agent Grounding

: Considers the executability of actions given the specification of the computer agent

Recursive Criticism and Improvement (RCI)

Agent의 입력으로 사용되기 위한 3단계의 reasoning

1. 주어진 Task를 풀기 위한 Outlined Plan을 생성

2. 그 Plan을 수행하기 위한 action들을 생성

3. Task를 수행하기 더 적합한 형태로 앞서 생성된 Action들을 improve
   (Task Grounding) Based on the current plan and task, the next proper instruction should be

4. 생성한 Action이, 현재 state에서 수행 가능한지 평가하고 improve
   (State Grounding) Considering the output on the webpage, the specific instruction should be

5. 생성한 Action이, 실행 가능한 형태가 될때까지 improve – format error 등 교정
   (Agent Grounding) This action does not match the regular expressions.
   The updated instruction that matches one of the regular expressions is

Experiments

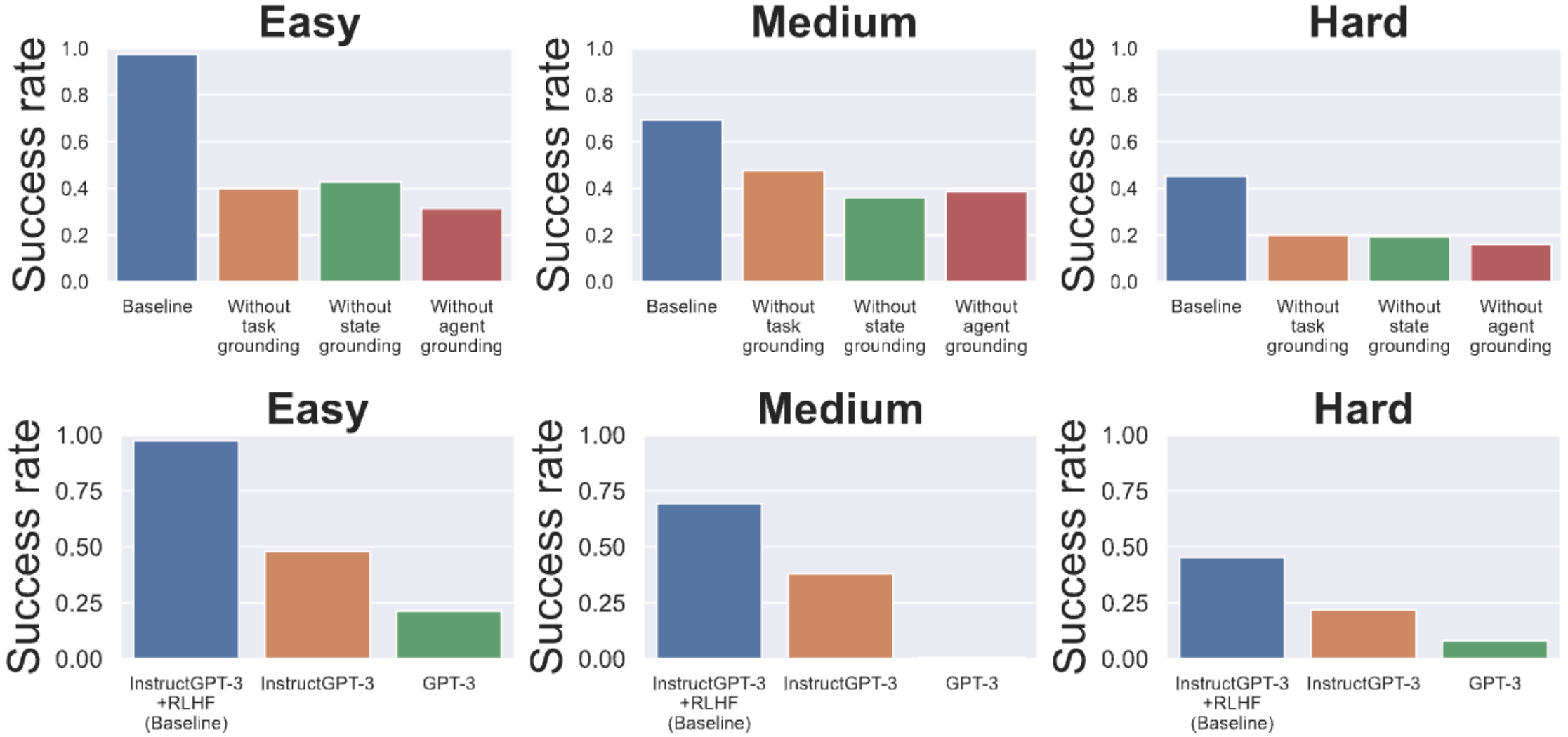
GPT3 기반의 실험

	Arithmetic						Common Sense	
	GSM8K	MultiArith	AddSub	SVAMP	SingleEq	AQuA	CommonSenseQA	StrategyQA
Zero-Shot	77.95	94.48	88.58	80.70	86.61	60.23	64.56	48.81
Zero-Shot + RCI	**85.43**	**97.64**	**89.76**	**84.65**	**94.49**	**67.32**	**68.11**	**61.81**

	GSM8K	MultiArith	AddSub	SVAMP	SingleEq
Zero-Shot	78.35	96.06	85.83	78.35	91.34
Zero-Shot + RCI	85.43	97.64	89.76	84.65	**94.49**
Zero-Shot CoT	82.28	96.85	83.86	79.92	89.37
Zero-Shot CoT + RCI	**86.22**	97.24	89.88	85.83	90.94
Few-Shot CoT	80.31	98.82	89.37	83.46	91.73
Few-Shot CoT + RCI	84.25	**99.21**	**90.55**	**87.40**	93.70

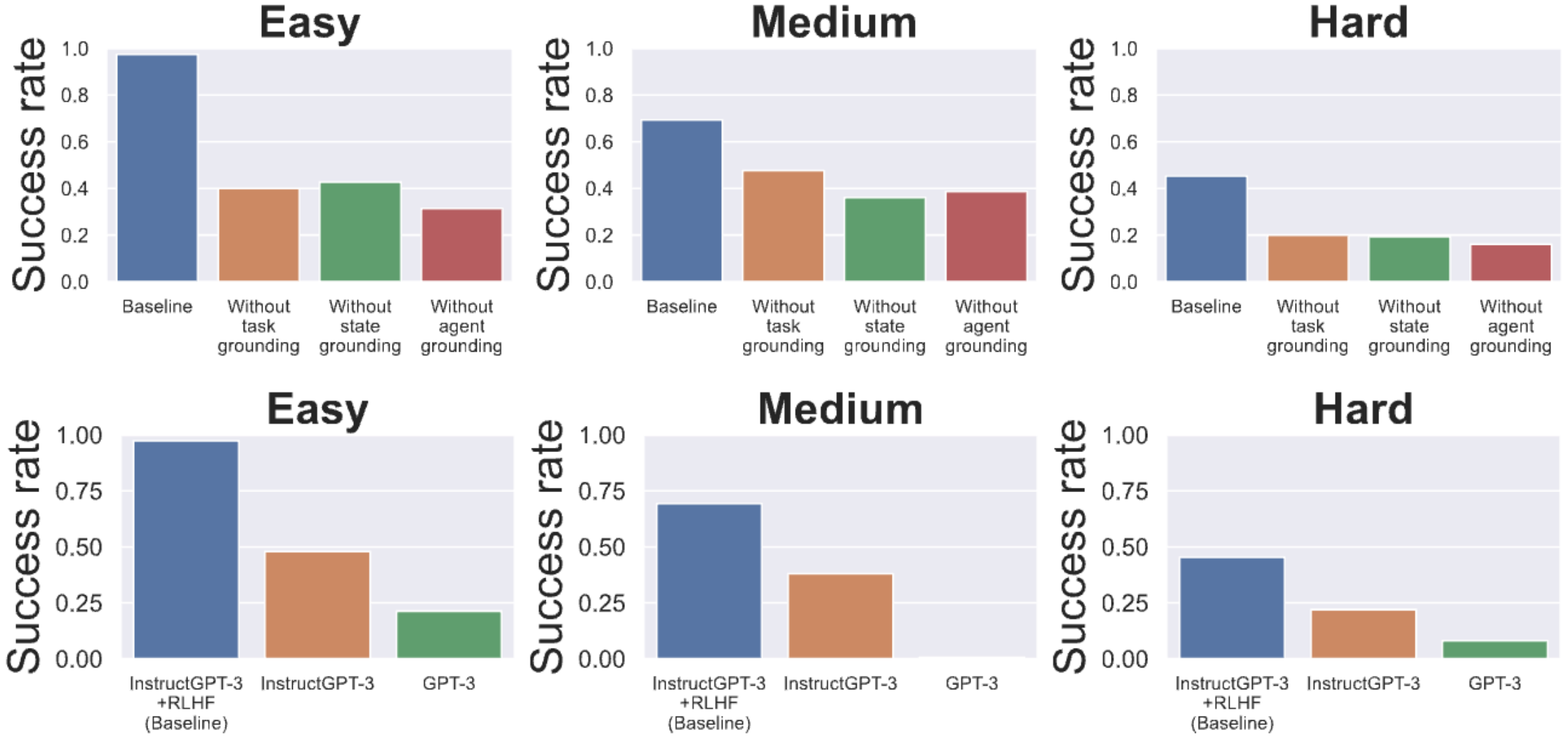
Experiments

GPT3 기반의 실험

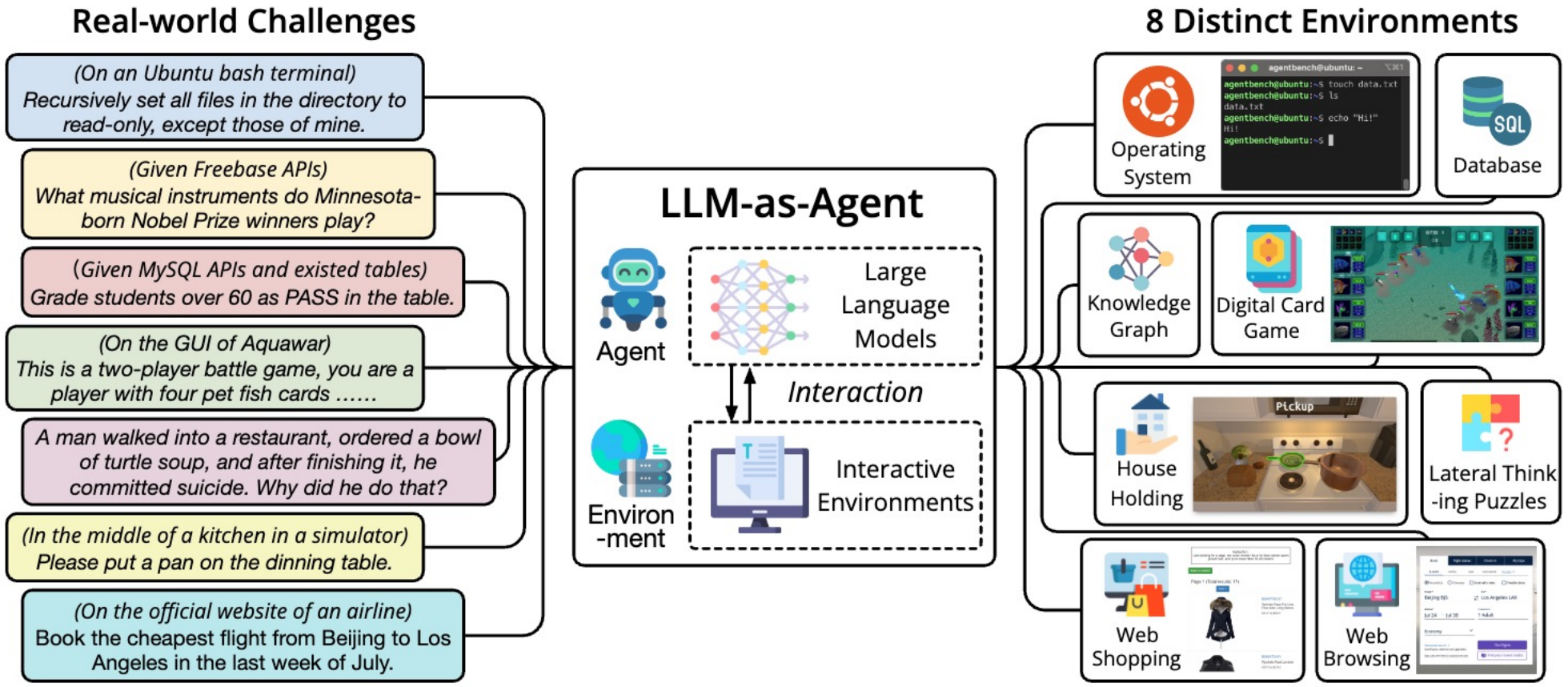


Experiments

GPT3 기반의 실험



이런류의 Agent들



LLM-as-Agent의 역할을, 여러 real-world tasks을 수행하는 보조도구로 간주

→ "각 LLM들은 Agent로서 얼마나 효과적인가"에 대한 종합적 평가

8가지의 Agent Task

- Code-grounded
  - Operating System
  - Database
  - Knowledge Graph
- Game-grounded
  - Digital Card Game
  - Lateral Thinking Puzzles
  - House-Holding
- Web-grounded
  - Web Shopping
  - Web Browsing

	Operating System	Data-Base	Knowledge Graph	Digital Card Game	Lateral Thinking Puzzle	House Holding	Web Shopping	Web Browsing
#Avg. Round	8	5	15	30	25	35	5	10
Metric	SR	SR	F1	Reward	Game Progress	SR	Reward	Step SR
#Dev	26 / 240	60 / 300	20 / 300	12 / 360	20 / 500	20 / 700	80 / 400	31 / 400
#Test	144 / 1200	300 / 1500	150 / 2250	20 / 600	50 / 1250	50 / 1750	200 / 1000	100 / 1000
Weight^{-1}	10.8	13.0	13.9	12.0	3.5	13.0	30.7	11.6

Code-grounded

Operating System

number of users with non-/home directories in an OS.

Action Space: Bash script

Natural Language	Bash Command(s)

find .java files in the current directory tree that contain the pattern 'TODO' and print their names	`grep -l "TODO" *.java`
	`find . -name "*.java" -exec grep -il "TODO" {} \;`
	`find . -name "*.java" | xargs -I {} grep -l "TODO" {}`

display the 5 largest files in the current directory and its sub-directories	`find . -type f | sort -nk 5,5 | tail -5`
	`du -a . | sort -rh | head -n5`
	`find . -type f -printf '%s %p\n' | sort -rn | head -n5`

search for all jpg images on the system and archive them to tar ball "images.tar"	`tar -cvf images.tar $(find / -type f -name *.jpg)`
	`tar -rvf images.tar $(find / -type f -name *.jpg)`
	`find / -type f -name "*.jpg" -exec tar -cvf images.tar {} \;`

Database

Table-Database 기반의 QA 수행

Action Space: SQL script

```
Question: the bipropellant rocket engine with kerosene and gas-generator
    cycle is designed by who?
Entities: [Gas-generator cycle, Kerosene]
```

Knowledge Graph

decision-making abilities of LLM

Action Space: Graph 탐색 행동

```
Action: get_relations(Gas-generator cycle)

Observation: [spaceflight.rocket_engine_cycle.rocket_engines, spaceflight
    .satellite.orbiting, spaceflight.rocket_stage.fuel, spaceflight.
    satellite_manufacturer.spacecraft_manufactured, automotive.fuel.
    engines]

Thought: From the above relations, I might use spaceflight.
    rocket_engine_cycle.rocket_engines to find the engines of Gas-
    generator cycle.
```

Code-grounded

Digital Card Game

The Best AI Gaming Platform

AI들끼리 경쟁하는 카드게임 승리전략

Action Space: 물고기 행동

- **Eel**
  - **Deflect (Passive)**: Distributes 70% damage to teammates and takes 30% when attacked. Gains 40 attack points after taking 200 damage accumulated.
  - **AOE (Active)**: Attacks all enemies for 35% of its attack points.
- **Sunfish**
  - **Deflect (Passive)**: Distributes 70% damage to teammates and takes 30% when attacked. Gains 40 attack points after taking 200 damage accumulated.
  - **Infight (Active)**: Inflicts 75 damage on one living teammate and increases your attack points by 140.

Lateral Thinking Puzzles

Yes/No형 질의응답의 반복으로 최종 답변 찾기

Action Space: Game master에게의 질문 (yes/no)

한 남자가, 어느 바닷가 레스토랑에서 바다거북 수프를 주문했으며 그 남자는 바다거북 수프를 한 수저 먹고는 주방장을 불렀다.
"죄송합니다. 이거 정말로 바다거북 수프인가요?"
"네, 틀림없는 바다거북 수프 맞습니다."
남자는 계산을 마친 뒤 집에 돌아가서 자살했다.
왜 그랬을까?

House-Holding

put the lamp on the table

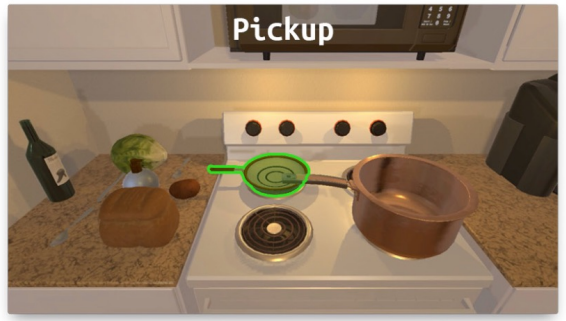
Action Space: 주방에서 할 수 있는 행동

```
> goto the cabinet

You arrive at the cabinet.
The cabinet is closed.

> open the cabinet

The cabinet is empty.
```



Typical Types of Finish Reason

Complete

Agent가 정상적으로 답변하는걸 성공하는 경우

Context Limit Exceeded (CLE):

the length of interaction history exceeds the LLM's maximum context length (only happened in 2,048-length LLMs text-davinci-002 and 003).

Invalid Format (IF):

the agent does not follow the format instruction.

Invalid Action (IA)

the agent follows the format instruction, but its selected action is invalid.

Task Limit Exceeded (TLE)

the agent does not solve the problem after reaching the predefined maximum interaction rounds or begins to do repeated generations for many rounds.

Experiments

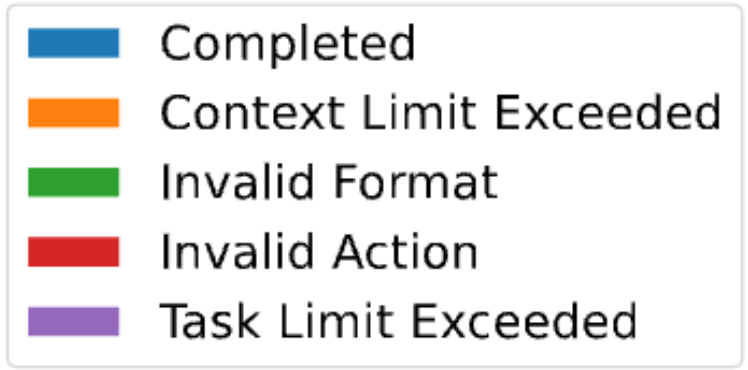
LLM Type	Models	VER	OA	Code-grounded			Game-grounded			Web-grounded	
				Operating System	Data-base	Know-ledge Graph	Digital Card Game	Lateral Thinking Puzzle	House Holding	Web Shopping	Web Browsing
API	gpt-4	0613	**4.01**	**42.4**	32.0	**58.8**	**74.5**	**16.6**	**78.0**	61.1	**29.0**
	claude-3	opus	3.11	22.9	**51.7**	34.6	44.5	14.3	70.0	27.9	26.0
	glm-4	-	2.89	29.2	42.3	46.3	34.1	14.2	34.0	61.6	27.0
	claude-2	-	2.49	18.1	27.3	41.3	55.5	8.4	54.0	61.4	0.0
	claude	v1.3	2.44	9.7	22.0	38.9	40.9	8.2	58.0	55.7	25.0
	gpt-3.5-turbo	0613	2.32	32.6	36.7	25.9	33.7	10.5	16.0	**64.1**	20.0
	text-davinci-003	-	1.71	20.1	16.3	34.9	3.0	7.1	20.0	61.7	26.0
	claude-instant	v1.1	1.60	16.7	18.0	20.8	5.9	12.6	30.0	49.7	4.0
	chat-bison-001	-	1.39	9.7	19.7	23.0	16.6	4.4	18.0	60.5	12.0
	text-davinci-002	-	1.25	8.3	16.7	41.5	11.8	0.5	16.0	56.3	9.0
OSS (Large)	llama-2-70b	chat	**0.78**	**9.7**	13.0	**8.0**	**21.3**	0.0	2.0	**5.6**	**19.0**
	guanaco-65b	-	0.54	8.3	**14.7**	1.9	0.1	**1.5**	**12.0**	0.9	10.0
OSS (Medium)	codellama-34b	instruct	**0.96**	2.8	**14.0**	**23.5**	8.4	0.7	4.0	**52.1**	**20.0**
	vicuna-33b	v1.3	0.73	**15.3**	11.0	1.2	**16.3**	1.0	**6.0**	23.9	7.0
	wizardlm-30b	v1.0	0.46	13.9	12.7	2.9	0.3	**1.8**	**6.0**	4.4	1.0
	guanaco-33b	-	0.39	11.1	9.3	3.2	0.3	0.0	**6.0**	6.2	5.0
OSS (Small)	vicuna-13b	v1.5	**0.93**	10.4	6.7	9.4	0.1	**8.0**	8.0	41.7	12.0
	llama-2-13b	chat	0.77	4.2	11.7	3.6	**26.4**	0.0	6.0	25.3	13.0
	openchat-13b	v3.2	0.70	**15.3**	12.3	5.5	0.1	0.0	0.0	**46.9**	**15.0**
	wizardlm-13b	v1.2	0.66	9.0	**12.7**	1.7	1.9	0.0	**10.0**	43.7	12.0
	vicuna-7b	v1.5	0.56	9.7	8.7	2.5	0.3	6.4	0.0	2.2	9.0
	codellama-13b	instruct	0.56	3.5	9.7	**10.4**	0.0	0.0	0.0	43.8	14.0
	codellama-7b	instruct	0.50	4.9	**12.7**	8.2	0.0	0.0	2.0	25.2	12.0
	koala-13b	-	0.34	3.5	5.0	0.4	0.1	4.4	0.0	3.9	7.0
	llama-2-7b	chat	0.34	4.2	8.0	2.1	6.9	0.0	0.0	11.6	7.0
	codegeex2-6b	-	0.27	1.4	0.0	4.8	0.3	0.0	0.0	20.9	11.0
	dolly-12b	v2	0.14	0.0	0.0	0.0	0.1	1.2	0.0	0.4	9.0
	chatglm-6b	v1.1	0.11	4.9	0.3	0.0	0.0	0.0	0.0	0.5	4.9
	oasst-12b	sft-4	0.03	1.4	0.0	0.0	0.0	0.0	0.0	0.3	1.0

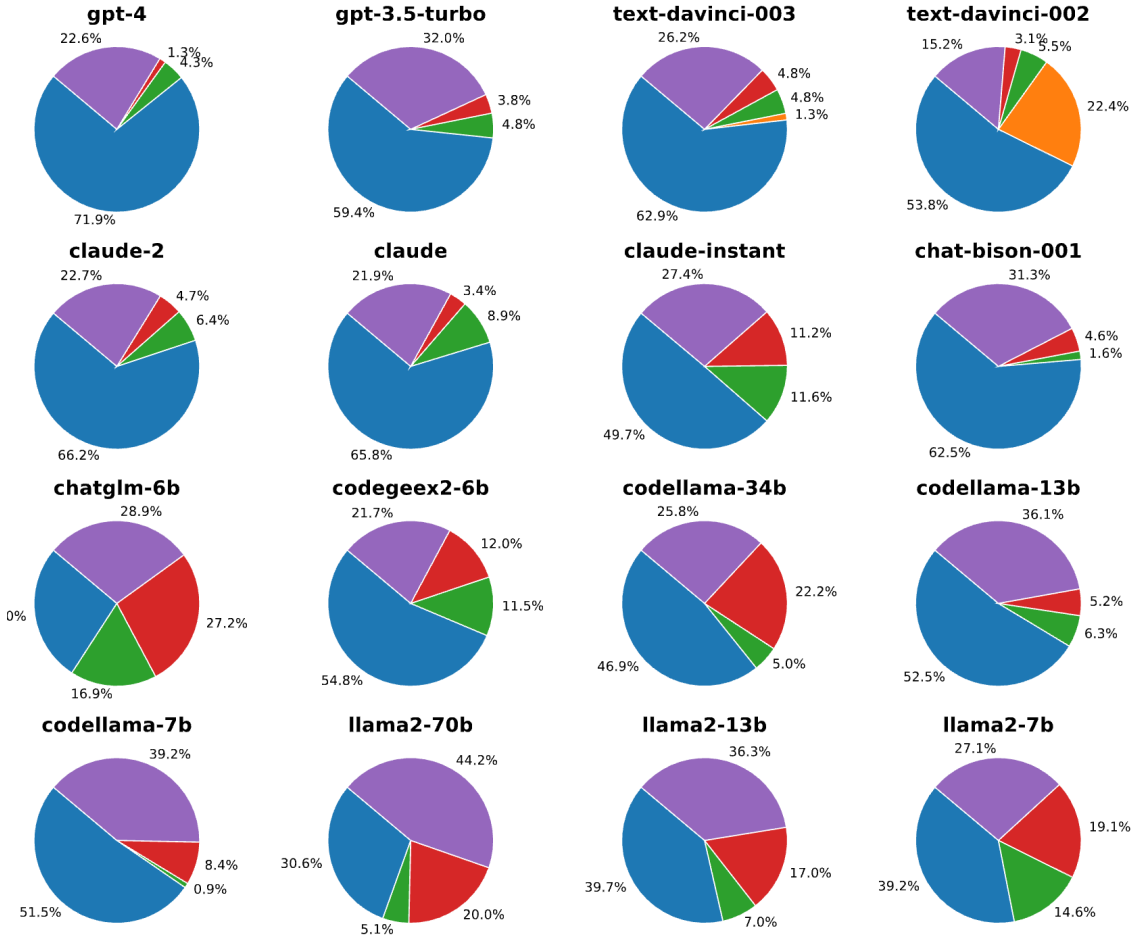
기본적으로 zero-shot COT 사용하여 평가

각 task 점수 기준: 0~100

Experiments

	Operating System	Database	Knowledge Graph	Digital Card Game	Lateral Thinking Puzzle	House Holding	Web Shopping	Web Browsing
Completed	75.0	37.9	30.1	51.2	14.0	13.1	54.9	56.6
CLE	0.1	0.7	2.0	0.0	3.5	0.7	0.0	0.0
Invalid Format	0.0	53.3	0.0	38.5	0.0	0.0	17.2	0.0
Invalid Action	0.9	0.0	0.0	10.2	0.0	64.1	0.0	8.4
TLE	23.9	8.0	67.9	0.0	82.5	22.1	27.8	35.0





대부분의 오류가 TLE에서 발생

Conclusion

- 반드시 Instruction Following이 전제되어야 가능한 LLM-as-Agent 연구

- Open source 모델에서는 Llama3로 와서야 겨우 높은 Instruction following 성능이 확보되고 있는 상황. 이제 앞으로 더 활발하게 연구될듯 함

- Test-Time scaling 등, 단순 "LLM 사용"이 아닌 "Agent 활용" 개념에서 적용될 수 있는 연구들도 개발되고 있는 상황 → 발전 가능성이 있으니.. 관심을 한번 가져보자

감사합니다