The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 24, No. 3, pp.87-94, Jun. 30, 2024. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2024.24.3.87
JIIBC 2024-3-13

대화형 텍스트 기반 게임에서 LLM의 게임플레이 기능 평가에 관한 연구

A Study on the Evaluation of LLM's Gameplay Capabilities in Interactive Text-Based Games

이동철*

Dongcheul Lee*

요 약 LLM(Large Language Model)을 활용하여 사전에 게임 데이터 학습 없이 텍스트 기반 게임을 수행할 수 있는지 알아보았다. LLM을 구현한 시스템으로는 ChatGPT-3.5와 가장 최신 형태인 ChatGPT-4를 채택하였다. 이에 더해 ChatGPT-4에 본 논문에서 제안하는 영구 메모리 기능을 추가하여 세 개의 게임 플레이어 에이전트를 제작하였다. 텍스트 기반 게임으로 가장 유명한 Zork를 활용하여 복잡한 장소를 이동해가며 정보를 모으고 퍼즐을 풀 수 있는지 알아보았다. 그 결과 세 에이전트 중 영구 메모리 기능을 추가한 에이전트의 성능이 탐험을 가장 넓은 범위로 진행하였고 점수도 가장 뛰어났다. 그러나 세 에이전트 모두 퍼즐을 푸는데 한계를 보였으며 이는 다단계 추론이 필요한 문제에 LLM이 취약하다는 것을 보여주었다. 그럼에도 여전히 본 논문에서 제안하는 에이전트를 사용하면 전체 장소의 37.3%를 방문하고, 방문했던 장소의 아이템을 모두 모으는데 성공할 수 있었던 것으로 LLM의 가능성을 확인할 수 있었다.

Abstract We investigated the feasibility of utilizing Large Language Models (LLMs) to perform text-based games without training on game data in advance. We adopted ChatGPT-3.5 and its state-of-the-art, ChatGPT-4, as the systems that implemented LLM. In addition, we added the persistent memory feature proposed in this paper to ChatGPT-4 to create three game player agents. We used Zork, one of the most famous text-based games, to see if the agents could navigate through complex locations, gather information, and solve puzzles. The results showed that the agent with persistent memory had the widest range of exploration and the best score among the three agents. However, all three agents were limited in solving puzzles, indicating that LLM is vulnerable to problems that require multi-level reasoning. Nevertheless, the proposed agent was still able to visit 37.3% of the total locations and collect all the items in the locations it visited, demonstrating the potential of LLM.

Keywords : Large Language Model, Text-based Games, ChatGPT, Zork

*종신회원, 한남대학교 멀티미디어공학과
접수일자 2024년 5월 22일, 수정완료 2024년 5월 30일
게재확정일자 2024년 6월 7일

Received: 22 May, 2024 / Revised: 30 May, 2024 /
Accepted: 7 June, 2024
*Corresponding Author: jackdclee@hnu.kr
Department of Multimedia Engineering, Hannam University, Korea

Ⅰ. 서 론

인공 지능 시스템이 자연어 처리 및 생성에 더욱 능숙해짐에 따라 사용자와 상호작용이 많은 게임 분야에서 다양하게 활용할 수 있는 가능성이 더 커졌다[1]. 특히 텍스트 기반 게임은 플레이어가 상세한 설명을 읽고 해석한 후 명령을 텍스트로 입력하여 게임을 진행해야 한다. 이러한 게임은 플레이어의 게임 내러티브 이해와 그에 따른 명령 입력에 크게 의존하기 때문에 사람과 유사한 텍스트를 처리하고 생성하도록 설계된 대규모 언어 모델 (Large Language Model, LLM)을 위한 좋은 시험장이 될 수 있다[2].

LLM은 사용자가 입력한 텍스트를 구문 분석하고 이에 대한 응답을 생성해 낼 수 있다. 이를 위해 별도의 사전 학습이나 튜닝이 필요하지 않으며, 사용자가 몇 가지 예시를 제공하여 대답을 사용자 의도대로 구조화할 수 있다. 이러한 능력으로 인해 연구자들은 LLM의 문제 해결 능력에 대해 활발히 연구하고 있으며, 특히 계획 및 추론 능력에 관심이 많다[3].

본 연구는 LLM이 텍스트 기반 게임에서 게임 환경을 이해하고 그에 따른 다음 명령을 생성할 수 있다는 가설에 근거한다. 또한 텍스트 기반 어드벤처 게임이 제공하는 복잡한 내러티브 및 의사 결정 기반 환경을 탐색하는 데 ChatGPT와 같은 LLM을 적용하는 방법을 고전 게임인 Zork에 초점을 맞춰 살펴볼 것이다.

이를 위해 Zork 게임 플레이에서 ChatGPT의 성능을 평가하여 게임 환경을 탐색하는 능력 및 게임 내 중요 아이템 수집 능력을 분석할 것이다. 또한 게임 능력을 향상시키기 위해 본 논문에서 제안하는 영구 메모리 기능을 추가하여 ChatGPT 만을 활용했을 경우와 성능을 비교 평가할 것이다.

본 연구로 인해 LLM의 계획 및 추론 능력에 대한 이해를 높일 수 있으며, 그 한계를 이해하고 발전시키기 위한 가능성을 발견할 수 있을 것이다. 이는 미묘한 텍스트의 이해와 내러티브에 맞는 적절한 응답 생성이 중요한 콘텐츠 제작, 교육, 엔터테이먼트, 심리 치료와 같은 분야에 특히 중요하다. 더 나아가 사회 문제에 대한 원인 분석 및 해결 방안 제시에 인공 지능이 점점 더 많이 활용될 수 있다는 중요한 의미를 갖는다.

Ⅱ. 관련 연구

텍스트 기반 게임은 플레이어와 상호작용 시 텍스트를 사용한다. 이러한 게임은 1980년대 초반에 주로 개발되었으며 Zork나 Colossal Cave Adventure와 같은 게임이 대표적이다. 플레이어는 환경, 캐릭터, 이벤트에 대한 설명을 읽고 텍스트 명령을 입력하여 게임을 진행한다.

이는 LLM의 동작 원리와 잘 맞아떨어지므로, 게임에서 생성된 환경 설명에 따라 LLM 기반 에이전트가 적절한 명령을 생성하는 방법에 대한 연구가 다양하게 진행되어왔다. [4]은 인간의 텍스트 기반 게임 플레이 데이터 세트를 사전에 학습하여, 이를 기반으로 상황에 맞는 액션을 생성하여 게임 플레이 효율성을 향상 시켰다. [5]는 게임 명령 생성을 위해 GPT-2와 BERT (Bidirectional Encoder Representations from Transformers) 모델을 결합한 아키텍처를 사용하는 것을 제안했다. 과거와 현재의 컨텍스트를 기반으로 미래 행동을 예측하기 위한 GPT-2와, 게임 상태와 플레이어의 질문을 처리하여 정확하고 관련성 있는 답변을 생성할 수 있는 BERT를 결합하여 텍스트 기반 게임에 적용함으로써 성능을 향상시켰다. [6]는 텍스트 기반 게임에서 행동을 예측하고, 게임 상태를 이해하며, 의사를 스스로 결정하기 위해 PLM (Pre-trained Language Models)을 활용할 것을 제안했다. 이 PLM을 통해 게임 데이터를 학습한 뒤, 상황에 적합하고 게임 진행에 효과적인 액션을 생성했다. 또한 동적 환경 모델링을 하고 게임 상태 그래프의 변화를 예측하여 전반적인 게임 플레이 성능을 향상시키고 상호작용 품질을 개선할 수 있었다.

또한 LLM의 한계를 지적한 연구도 다양하게 진행되었다. 일반적으로 LLM은 자연어 이해 및 생성에 있어 뛰어나지만, 논리적 일관성, 다단계 추론, 문맥 인식, 실시간 계획 등의 분야에는 취약한 것으로 알려져 있다. [7]에서는 LLM은 복잡한 문제를 추론해야 할 때 논리적으로 일관되지 않은 결과를 산출하는 경우가 많으며, 이로 인하여 정확하고 엄격한 논리적 추론이 필요한 작업에서 신뢰성을 보장할 수 없다고 했다. [8]에서는 LLM은 기본 개념에 대한 깊은 이해보다는 주로 패턴 인식을 통해 작동하기 때문에, 이전에 학습했던 익숙한 상황에서는 올바른 응답을 생성할 수 있지만, 더 깊은 이해가 필요한 새로운 상황에서는 실패할 수 있다고 하였다. [9]에서는 LLM은 다단계 추론에 어려움을 겪으며, 복잡한 문제를 해결하는 데 필요한 논리적 순서를 놓치는 경우가 많기 때문에, 순차적인 의사 결정

과 장시간의 추론이 필요한 작업에서 효율성이 떨어진다고 하였다. [10]는 다양하고 복잡한 게임 공간을 관리하고 최적화하는 것은 LLM에게 어려우므로 최적화된 계획과 의사 결정을 하는데 효과적이지 않다고 하였다.

Ⅲ. 배경 및 환경 설정

1. ChatGPT

ChatGPT는 OpenAI에서 LLM을 기반으로 개발한 대화형 인공 지능 모델이다. 이는 사람이 생성하는 텍스트와 유사한 텍스트를 생성하도록 설계된 일종의 머신 러닝 모델인 GPT (Generative Pre-trained Transformer) 아키텍처를 갖는다. 여기서 Generative는 모델이 텍스트를 생성한다는 의미이며, Pre-trained는 특정 작업에 맞게 미세 조정되기 전에 대규모 텍스트 말뭉치에 대해 미리 학습되었다는 것을 의미한다. 또한 Transformer는 텍스트와 같은 데이터 시퀀스를 처리하는 데 효과적인 신경망 아키텍처를 가지는 것을 의미한다. ChatGPT는 다양하고 방대한 텍스트 데이터로 학습되며, 이 데이터의 패턴과 정보를 사용하여 질문에 대한 응답을 생성한다. 하지만 다른 인공 지능 모델과 마찬가지로 한계가 있으며 항상 완벽하거나 사실에 근거한 정확한 답변을 제공하지는 못할 수도 있다.

현재 서비스되고 있는 ChatGPT의 버전은 3.5와 4.0으로 나뉜다. 이 둘은 기능, 학습 데이터 및 전반적인 성능에서 차이가 있다. ChatGPT-3.5는 GPT-3에 기반하고 있으며 2021년까지 책, 웹사이트 및 기타 텍스트가 포함된 데이터 세트에서 데이터를 학습하였다. 입력된 질문을 기반으로 문맥을 이해하고 답변을 생성하는데 우수하며 광범위한 주제를 처리할 수 있다. ChatGPT-4는 더욱 발전된 GPT-4를 기반으로 하며, 복잡한 질문에 더 적절히 대답하며 부정확한 답변을 생성할 가능성이 줄었다. 또한, 길게 이어지는 대화에서 보다 일관성 있는 답변을 제공하고, 이전의 맥락을 잘 이해할 수 있다. 뿐만 아니라, 더 크고 다양한 데이터 세트에 대해서 최신 정보를 포함하여 학습하였기 때문에 최신 주제에 대해 더 잘 답변할 수 있다.

2. Zork

Zork는 1977년에 MIT에서 개발된 텍스트 기반 어드벤처 게임이다. 그 이후 1980년 Infocom에서 상용화되었고, 당시 개인용 컴퓨터의 메모리 한계로 인해 Zork I 위대한 지하 제국, Zork II 프로보즈의 마법사, Zork III 던전 마스터로 나누어졌다. 플레이어는 캐릭터에게 텍스트 명령을 입력하여 장소를 이동하고, 퍼즐을 풀고, 보물을 수집한다. 게임에는 각각 이름과 설명이 있는 수백 개의 장소가 있으며, 플레이어의 명령은 플레이어가 위치한 장소와 상호작용한다. 명령은 장소의 문맥에 맞아야 하며, 문맥에 맞지 않을 경우 게임은 플레이어에게 명령을 다시 입력하도록 요청한다. 게임은 플레이어의 현재 위치, 플레이어 행동에 대한 결과, 기타 환경 요소에 대한 설명을 텍스트로 사용자에게 응답한다.

게임의 목표는 게임 내 모든 보물을 수집하는 것이며, 보물을 수집하거나 임무를 완수할 때마다 플레이어의 점수가 증가한다. 플레이어는 자유롭게 게임 세계를 이동하며 보물을 수집할 수 있지만, 일부 통로는 통과하기 위해 퍼즐을 푸는 것이 필요하다. 예를 들어 Dome Room에서 Torch Room으로 내려가기 위해서는 난간에 밧줄을 묶어야 한다. 또한, 밧줄 끝이 바닥에서 10피트 떨어져 있기 때문에 내려갈 수는 있지만, 다시 올라갈 수는 없는 제약이 있다. Loud Room은 플레이어가 행동을 수행하기에는 너무 시끄러우므로 근처 댐의 물을 비워 물 떨어지는 소리를 멈추게 하거나, 방에서 “Echo”를 외쳐 방의 음향을 바꿔야 한다. 어두운 지역에서는 플레이어가 손전등을 켜고 있어야 몬스터에게 공격당하지 않는다. 가지고 다닐 수 있는 물건의 수량에는 제한이 없으나 총 무게에는 제한이 있으므로 필요 없는 무거운 물건은 버리는 것이 좋다.

3. 실험 환경

실험 환경은 그림 1과 같이 Zork I 실행 파일과 Chat Completions API, 그리고 이 둘을 연동해주고 상태를 관리해 줄 Python 프로그램으로 구성되어 있다.

Zork I의 최초 소스 코드는 MDL (Model Development Language)로 작성되었고, 이후에 FORTRAN으로 재작성되었으며, 나중에 C로 변환되었다. 본 실험에서 사용한 Zork I 소스 코드는 Infocom에서 공개 도매인에 C언어로 공개한 것을 수정 및 컴파일하여 사용하였다[11].

Python 프로그램에서는 subprocess module을 통해 컴파일된 Zork I을 실행시켰으며, input stream과 output stream을 통해 플레이어 명령을 전송하고, 게

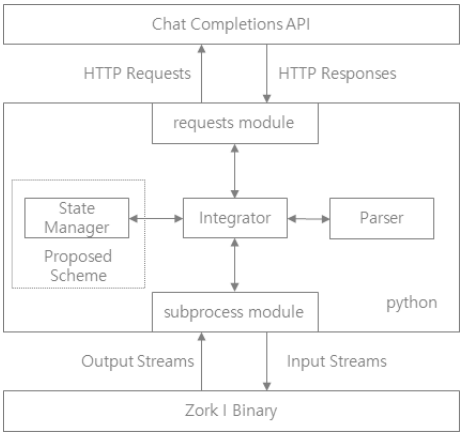

그림 1. 실험 환경에서 사용된 시스템 구조
Fig. 1. System architecture used in the experiment environment

```
{"role": "system", "content":
"You are a PLAYER in a text adventure game. I will provide the text that the game would have generated, and you must respond with what actions the PLAYER would take.
PLAYER must use following commands only: go north south east west northeast northwest southeast southwest up down climb look enter in out get throw open close read drop put turn move attack examine inventory eat shout tie pick break kill pray drink smell cut listen echo.
PLAYER must only respond with 1 to 5 words.
PLAYER must always use the format \"〈Command〉 〈Other words〉\" when responding.
PLAYER must use \"Look around\" to figure out what PLAYER can do, If the text has negative messages like \"I don't understand,\" \"You can't,\" \"That's not,\" or \"You don't,\". Never ask me what to do.
Never apologize.
Never provide explanations.
Never respond with anything at all other than the command."}
```

그림 2. Chat Completions API에 사용된 시스템 메시지
Fig. 2. System message used in the Chat Completions API

임의 응답을 받았다. 게임의 응답을 Chat Completions API로 직접 전달하기에는 불필요한 정보들이 섞여 있기 때문에 Parser를 통해 필요한 정보만 추려내었다. Parser에서 추려내는 정보에는 현재 플레이어의 장소, 점수, 움직임 수, 응답 메시지가 있다. Integrator는 Parser에게 정제된 정보를 받아 request module을 통해 HTTP Requests 메시지로 Chat Completions API를 호출하여 정보를 전달한다. HTTP Responses 메시지로 온 ChatGPT의 응답은 Integrator를 통해 Zork I 프로세스로 전달된다.

ChatGPT의 응답이 Zork I을 플레이하는데 사용될 명령이 되도록 하기 위해 API의 시스템 메시지를 그림 2과 같이 설정하였다. 우선 Zork에서 사용 가능한 명령어 집합을 [12]를 참고하여 설정하였다. 이러한 접근 방식은 응답을 간소화하고 ChatGPT가 게임 명령을 생성하기 위한 작업에서 너무 많이 벗어나지 않도록 집중력을 유지하고, 게임에서 이해할 수 없는 명령어를 사용하여 불필요한 오류 메시지를 발생시키는 것을 줄이기 위함이다. 또한, ChatGPT가 게임 명령을 직접 응답하는 것이 아니라, 게임 상황에 대한 설명을 하거나 사용자에게 어떻게 해야 한다고 알려주는 경우를 방지하기 위한 설정도 추가하였다. 뿐만 아니라, 게임의 오류 메시지에 대해서 ChatGPT가 과도하게 집중하거나 관련 없는 응답을 하는 것을 막기 위해 오류 메시지가 발생할 경우 주변을 둘러보고 무엇을 할 수 있는지 찾아보도록 설정하였다.

4. 성능 향상 방안

매번 게임 명령어를 입력할 때마다 ChatGPT와 인간이 다른 점 중의 하나는 그동안 지나왔던 게임 환경에 대한 기억을 인간은 더 잘한다는 것이다. ChatGPT는 상호작용 전반에 걸쳐 영구적인 메모리를 가지고 있지 않다. 대화 내의 단기적인 맥락은 처리할 수 있지만, 세션이 종료되면 과거의 상호작용 내역을 기억할 수 없다. 따라서 Zork처럼 복잡한 게임 맵에서 플레이어의 진행 상태를 추적하고 이전에 방문한 위치를 기억하기 어렵다. 그래서 ChatGPT가 인간과 비슷하게 게임을 할 수 있도록 Python 프로그램 내에 State Manager

(a)
```
West of House - Go North - North of House, North of House - Go North - Forest Path,
North of House - Go East - Behind House, Behind House - Enter - Kitchen
```

(b)
```
West of House -- North --> North of House --+
                                            |
                                            +-- North --> Forest Path
                                            |
                                            +-- East --> Behind House -- Enter --> Kitchen
```

그림 3. State Manager가 저장했던 플레이어의 경로를 ChatGPT가 해석하는 방식
Fig. 3. How ChatGPT interprets the player's path saved by State Manager

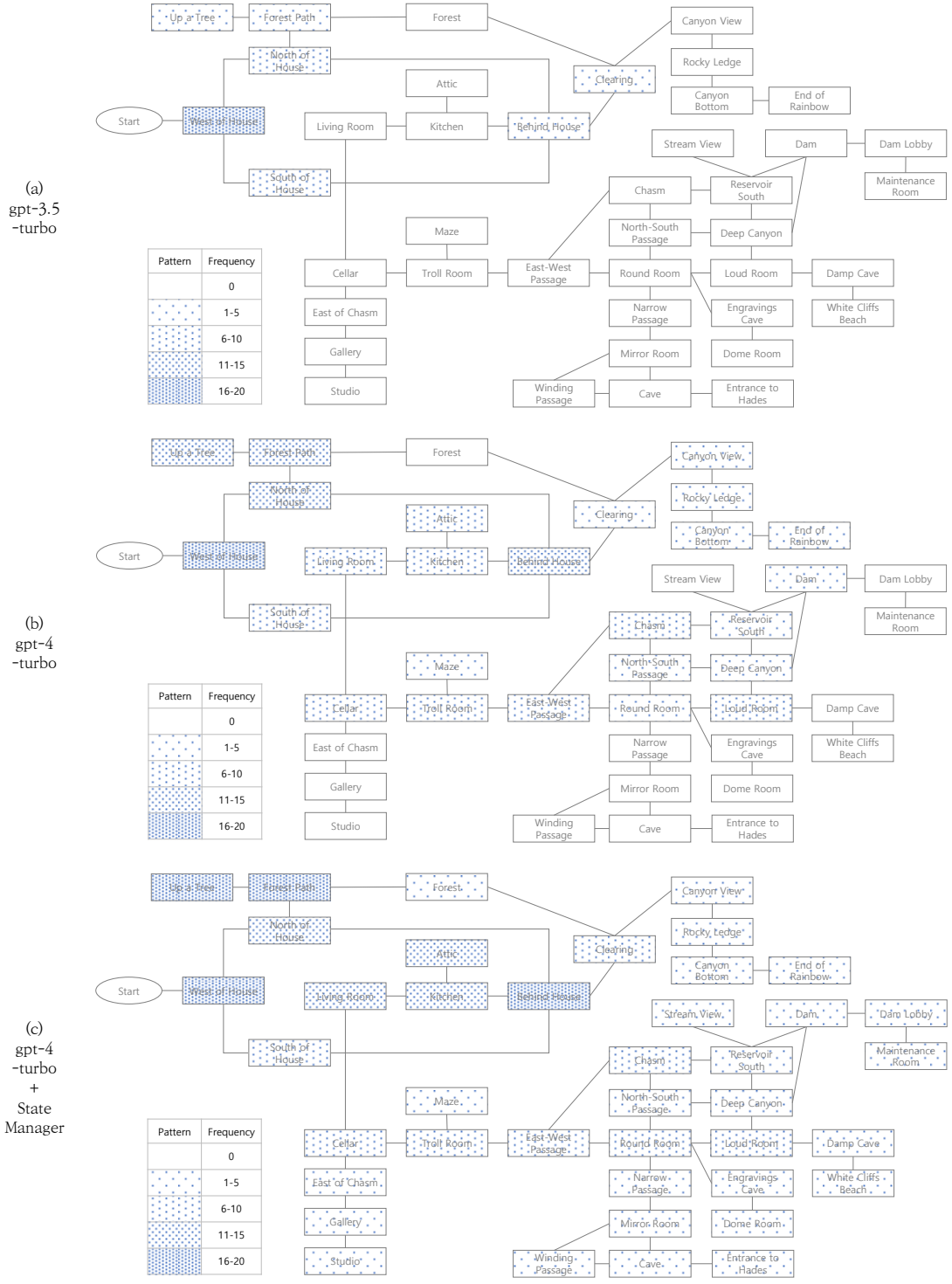

그림 4. 성능 평가에 사용된 에이전트가 각 장소를 방문한 빈도
Fig. 4. How often the agents used in the performance evaluation visited each place

표 1. 각 에이전트의 성능 실험 결과
Table 1. Performance evaluation results for the agents

Agent	Score	Moves	Deaths	Loops	Wrong commands	Finish
gpt-3.5-turbo	1.25	63.2	0	15	5	N
gpt-4-turbo	20.25	130.1	2	18	0	N
gpt-4-turbo + State Manager	28.4	193.5	2	18	0	N

를 추가하였다. State Manager는 게임 내 방문했던 위치를 기억하고 있다가 ChatGPT에게 게임 명령 생성 요청을 보낼 때 이 정보를 함께 제공하는 역할을 한다. State Manager는 출발지, 도착지, 이동 명령어 형태로 정보를 저장한다. 예를 들어 지나온 경로가 그림 3a와 같이 저장되었다면 ChatGPT는 이 정보를 통해 그림 3b와 같이 경로를 그려낼 수 있다. 또한, 이 정보를 통해 West of House에서 Kitchen으로 가는 방법을 물어보면 NORTH, EAST, ENTER와 같이 답을 해 줄 수 있다. State Manager는 ChatGPT의 이러한 추론 능력을 통해 인간의 기억력과 비슷한 능력을 제공해 줄 수 있도록 한다.

5. 실험 방법

ChatGPT가 Zork를 얼마나 잘 플레이할 수 있는지 알아보기 위해 세 가지 에이전트를 통해 성능을 평가하였다. 첫 번째 방법은 gpt-3.5-turbo 모델을 사용한 것, 두 번째 방법은 gpt-4-turbo 모델을 사용한 것, 마지막 방법은 gpt-4-turbo 모델 및 State Manager를 사용한 방법이다.

각 에이전트의 성능을 평가하기 위해 각 에이전트 당 20회의 게임을 실행하였다. 게임 종료 조건은 게임 최고 점수 350에 도달하거나, 게임 명령이 아닌 다른 내용을 답하여 게임을 진행할 수 없는 경우, 같은 장소에 연속적으로 20회 이상 계속 머물러 게임이 진행되지 않는 경우, 그리고 사망하였을 경우로 설정하였다. 평가 지표는 총 점수, 각 장소 방문 빈도로 정하였다. 총 점수는 보물을 얼마나 많이 모았는지를 측정할 수 있으며, 각 장소 방문 빈도는 어떤 방법을 사용하느냐에 따라 얼마나 다양한 장소를 탐험할 수 있는지를 측정할 수 있다.

Ⅳ. 실험 결과

각 에이전트가 게임 장소를 탐험할 때 얼마나 다양한 장소를 방문하는지를 평가하기 위해 에이전트별로 게임을 할 때마다 각 장소에 방문했던 건수를 산정해 보았다. 이 건수를 모두 합한 것을 지도에 나타낸 것이 그림 4이다. 한 게임 안에서 에이전트가 같은 장소를 반복해서 방문한 건수는 한 번으로 간주하였다. 원래의 Zork I의 지도는 이 지도보다 훨씬 크나, 세 에이전트가 한 번이라도 방문했던 장소만 그림 4에서는 나타내었다. 에이전트가 자주 방문한 장소일수록 장소의 색깔을 진하게 표시하였다.

그림 4(a)는 gpt-3.5-turbo를 사용한 에이전트의 결과를 나타낸다. 이 에이전트는 세 에이전트 중 가장 적게 게임 환경을 탐험하였다. 그 이유는 게임 시작점인 West of House에서 편지를 읽고 다음 장소로 이동을 못 한 경우가 많았고, 다음 장소로 이동을 했더라도 집 내부로 들어가지 못했기 때문이다. 집 내부로 들어가기 위해서는 창문을 여는 과제에 성공해야 하지만 이 과제를 한 건도 성공하지 못했다.

그림 4(b)와 (c)는 각각 gpt-4-turbo만 사용한 것과 State Manager를 함께 사용한 결과이다. 두 에이전트의 공통점은 Reservoir South에서 Reservoir를 가기 위해 다른 장소에 가서 저수지 물을 비우는 퍼즐을 풀어야 하는 것처럼, 특정 장소에 가기 위한 퍼즐이 존재할 경우 퍼즐을 하나도 풀지 못했다는 점이다. 차이점은 (c)의 경우 퍼즐이 존재하여 통과하지 못하는 장소까지 최대한 탐험할 수 있었던 반면, (b)의 경우 (c)가 방문했던 장소의 63%만 탐험할 수 있었다. Zork I에는 모두 110개의 장소가 있는데 각 에이전트 별로 전체의 6.4%, 23.6%, 37.3%를 탐험한 것이다. 각 장소를 탐험한 빈도도 (c)의 경우가 더 많았었는데, 이는 State Manager가 gpt-4-turbo 모델에 제공하는 정보가 게임 환경을 탐험하는 데 중요한 도움을 준다는 것을 의미한다.

표 1은 각 에이전트의 성능 실험 결과를 정량적으로 나타낸 표이다. gpt-3.5-turbo의 경우 평균 점수가 1.25로 매우 낮았는데, 이는 집 내부에 들어가지 못해 대부분의 게임 점수가 0이었으나 Egg 보물을 가끔 얻

는 경우가 있었기 때문이다. State Manager를 사용했던 에이전트와 사용하지 않았던 에이전트의 점수 차는 8.15였는데 이는 점수를 얻을 수 있는 장소인 Up a Tree, Kitchen, Cellar, The Troll Room, Loud Room과 같은 장소를 더 많이 방문했기 때문이다. 더 놀라운 점은 Gallery를 방문하여 보물을 수집한 것인데, 이는 State Manager를 사용하지 않았던 에이전트는 한 번도 방문하지 못했던 장소이다. 아쉬운 점은 보물을 모은 후 Living Room에 있는 trophy case 안에 넣어야 더 많은 점수를 얻을 수 있는데 이를 알지 못해 점수를 적게 받았다는 점이다. Moves의 경우 장소를 이동할 때마다 1씩 올라가는데 gpt-3.5-turbo의 경우 종료 조건이 빨리 왔기 때문에 Moves도 적게 나온 것을 알 수 있다. Deaths의 경우 The Troll Room에서 Troll과 싸우다가 죽는 것처럼 플레이어가 사망하였을 때의 횟수이다. gpt-3.5-turbo의 경우 사망할 가능성이 있는 장소에 가지도 못했기 때문에 횟수가 0이었다. Loops와 Wrong commands는 각각 같은 장소에 연속적으로 20회 이상 계속 머무를 경우와 사용 가능한 명령어가 아닌 게임 상황에 대한 설명만 하여 게임을 진행할 수 없는 경우의 수이다. gpt-3.5-turbo의 경우 시스템 메시지로 게임 설명을 하지 말라고 지시했음에도 불구하고 5번 발생했다. 세 에이전트 모두 대부분의 게임은 Loops 상황에서 종료되었으며 상황과 상관이 없는 한 가지 동일한 명령어를 계속 반복하거나, 무의미한 4~6개의 명령어를 반복적으로 실행하는 모습을 보였다. 특히 퍼즐이 존재하여 다음 장소로 이동할 수 없는 장소에서 이러한 현상이 자주 발생하였다. 아쉽게도 세 모델 모두, 보물을 전부 모아 350점으로 게임에 성공한 경우는 없었다.

V. 한 계

LLM은 주로 깊은 이해력보다는 패턴 인식에 의존한다. 따라서 패턴을 익힌 익숙한 상황에서는 적절한 답변을 생성할 수 있지만, 더 깊은 이해가 필요한 새로운 상황에서는 엉뚱한 답을 생성할 수도 있다. ChatGPT의 경우 Zork 게임을 전문적으로 학습한 모델이 아니기 때문에 실험에서 환경과 상관없는 대답을 하는 경우가 종종 발생하였다. 이러한 점은 사전에 학습이 필요한 강화 학습에 비해 성능이 많이 떨어지는 점이다.

또한, Zork에서는 퍼즐을 풀고 내러티브를 진행하기 위해 여러 단계의 추론이 필요한 경우가 많다. 그러나 LLM은 이러한 작업에 필요한 논리적 순서를 유지하지 못하는 경우가 많다. 게임 상태를 파악하지 못하거나 이전에 했던 행동들을 현재 상황과 인과 관계를 추론하지 못해 퍼즐을 풀어야 갈 수 있는 장소로 못 간 경우가 많았다.

뿐만 아니라, LLM은 상호작용 전반에 걸쳐 메모리를 유지하는 기능이 없다. 각 질문과 응답은 독립적으로 처리되기 때문에 과거 지식을 기반으로 하는 텍스트 기반 게임에서는 큰 단점이 된다. Zork에서는 플레이어가 각 장소의 위치, 아이템, 단서를 기억해야 한다. ChatGPT의 경우 여러 세션에 걸쳐 이러한 정보를 유지하기 어렵기 때문에 게임을 효과적으로 플레이할 수 없었다. 그나마 State Manager를 보조적으로 사용하면 영구 메모리 기능을 부분적으로 사용할 수 있기 때문에 더 나은 결과를 보여주었다.

VI. 결 론

본 논문에서는 LLM을 이용하여 텍스트 기반 게임을 플레이할 수 있는지 알아보았다. 이를 위해 ChatGPT-3.5 및 ChatGPT-4가 Zork 게임을 잘 플레이할 수 있는지 비교하였고, 영구 메모리 기능을 부분적으로 부여하기 위해 State Manager 기능을 추가하여 비교해 보았다. 그 결과 ChatGPT-3.5, ChatGPT-4, 그리고 State Manager를 추가한 에이전트 순서로 게임 환경을 더 잘 탐험하였고 미션도 잘 수행하여 점수도 높았다. 그러나 한계도 명확했는데 퍼즐을 풀어야 갈 수 있는 장소에는 어떤 에이전트도 가지 못했으며 상황과 맞지 않게 같은 명령어를 반복적으로 응답하는 경우도 많았다. 그러나 현실 세계를 반영하는 게임에서 사전에 아무런 게임 데이터 학습 없이 게임을 진행할 수 있었다는 점은 LLM이 우리 현실 세계 문제를 해결하는데 무한한 가능성을 가지고 있다는 점을 시사한다.

향후 연구로는 LLM이 가진 한계인 다단계 추론 능력 및 영구 메모리 능력을 향상시켜 사전 데이터 학습 없이 텍스트 기반 게임을 더 잘 플레이하도록 개선하는 것이다.

References

[1] S. Lim and S. Youn, "A Study on Efficient Natural Language Processing Method based on Transformer," The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 23, No. 4, pp.115-119, 2023. DOI: https://doi.org/10.7236/JIIBC.2023.23.4.115

[2] J. Lee and S. Kim, "Development of a Text-based Personal Content Creation System using Deep Learning," The Journal of Korean Institute of Information Technology, Vol. 21, No. 12, pp. 67-75, 2023. DOI: https://doi.org/10.14801/jkiit.2023.21.12.67

[3] Z. Zhang and J. Ahn, "Effect Influencing the Labor Market by Evolution of the ChatGPT as a Large-Scale Language Model," Journal of the Korea Academia-Industrial cooperation Society, Vol. 25, No. 4, pp. 227-235, 2024. DOI: https://doi.org/10.5762/KAIS.2024.25.4.227

[4] S. Yao, R. Rao, M. J. Hausknecht, and K. Narasimhan, "Keep CALM and Explore: Language Models for Action Generation in Text-based Games," Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 8736-8754, 2020. DOI: https://doi.org/10.48550/arXiv.2010.02903

[5] G. Furman, E. Toledo, J. Shock, and J. Buys, "A Sequence Modelling Approach to Question Answering in Text-Based Games," Wordplay and NLP: The 10th Workshop on Computational Approaches to Linguistic Creativity, 2022.

[6] M. Kim, Y. Jung, D. Lee, and S. Hwang, "PLM-based World Models for Text-based Games," Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022. DOI: https://doi.org/10.18653/v1/2022.emnlp-main.86

[7] K. Stechly, K. Valmeekam, and S. Kambhampati, "On the Self-Verification Limitations of Large Language Models on Reasoning and Planning Tasks," arXiv:2402.08115, 2024. DOI: https://dx.doi.org/10.48550/arXiv.2402.08115

[8] K. Valmeekam, S. Sreedharan, M. Niraula, and S. Kambhampati, "PlanBench: An Extensible Benchmark for Evaluating Large Language Models on Planning and Reasoning about Change," Proceedings of the Advances in Neural Information Processing Systems 36, 2022. DOI: https://doi.org/10.48550/arXiv.2206.10498

[9] Y. Xie, M. Liu, R.K. Vinayak, and N.R. Ahmed, "Translating Natural Language to Planning Goals with Large-Language Models," arXiv:2302.05128, 2023. DOI: https://dx.doi.org/10.48550/arXiv.2302.05128

[10] Z. Shi, Y. Xu, M. Fang, and L. Chen, "Self-imitation Learning for Action Generation in Text-based Games," Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pp. 703-726, 2023. DOI: https://dx.doi.org/10.18653/v1/2023.eacl-main.50

[11] https://github.com/devshane/zork

[12] https://zork.fandom.com/wiki/Command_List

저 자 소 개

이 동 철(종신회원)

- 2002년 : POSTECH 컴퓨터공학 학사
- 2004년 : POSTECH 전자컴퓨터공학 석사
- 2004년 ~ 2012년 : KT 중앙연구소 책임연구원
- 2012년 : 한양대학교 전자컴퓨터 통신공학 박사
- 2012년 ~ 현재 : 한남대학교 멀티미디어공학과 교수
- 관심분야 : 딥러닝, 자율주행, 신경망, 알고리즘

※ 이 논문은 2023학년도 한남대학교 학술연구비 지원에 의하여 연구되었음