LLM과 상호작용하는 RAG 기반 AI 캐릭터

신지호, 이세영, 최명재, 정은서, 이병정

서울시립대학교 컴퓨터과학부

sjm010529@gmali.com, ocdee39@gmail.com, ssy07124@uos.ac.kr, eunseoj@uos.ac.kr, bjlee@uos.ac.kr

A RAG based AI Character Interacting with a LLM

Jiho Shin, Seyeong Lee, Myoungjae Choi, Eunseo Jung, Byungjeong Lee

Dept. of Computer Science and Engineering, University of Seoul

요 약

본 논문은 인간과 일관적이고 자연스러운 대화가 가능한 LLM 기반 AI 캐릭터 시스템 구현을 설명한다. AI 캐릭터 시스템을 구현하기 위해 관련 연구를 참고하여 RAG 서브시스템을 설계하였다. 이때 기억을 장기 기억, 단기 기억으로 저장하여 대화의 흐름이 일관성 있도록 하기 위한 세 가지 특성을 정의하였다. 몇 가지 시나리오를 가정하여 사용자별로 각기 다른 대화를 진행하여 사용자와의 자연스러운 대화가 가능한 것을 확인하였다. 또한 LLM과의 상호작용 토큰 수 실험에서 모든 대화를 프롬프트에 컨텍스트로 추가한 방식에 비해 입력 토큰을 20% 적게 사용했다. 본 AI 캐릭터 시스템을 통하여 맞춤화된 경험을 제공하고 입력 토큰을 줄여 비용 절감 효과를 얻을 수 있다.

Ⅰ. 서 론

본 논문에서는 일관적이고 자연스러운 대화가 가능한 LLM(Large Languae Model) 기반 AI 캐릭터 시스템 구현을 설명한다. AI 캐릭터는 게임, 소설, 영화 등 다양한 엔터테인먼트 분야에서 활용되거나 파생될 수 있는데, 현재 상용화된 챗 서비스로 이를 온전히 구현하는 데에는 한계가 있다. 대표적인 예로, ChatGPT는 단순 챗 서비스로서 특정 성격이나 고유한 배경을 갖지 않는다. 따라서 사용자의 입력에 일률적인 답변을 보이며 주로 작업 보조 도구로 사용된다. 물론 특정 작품의 캐릭터를 흉내 내도록 역할극을 맡기거나, character.ai와 같은 커스텀 챗 서비스를 이용할 수도 있다[1]. 하지만 커스텀을 통해 구현된 챗봇은 원작 캐릭터를 완벽히 모방하지 못하고 어색한 답변을 생성하며, 이는 사용자의 흥미를 떨어뜨리는 요소로 작용한다. 본 논문에서 설명하는 AI 캐릭터 시스템은 차별된 설계를 통해 이러한 문제들을 해결한다. 기억 시스템을 구축하여, AI 캐릭터가 사용자와의 대화와 기본 상식 정보를 효율적으로 관리하고 답변에 반영할 수 있도록 한다. 이를 통해 일관적이고 자연스러운 대화가 가능하다.

Ⅱ. 관련 연구

게임 에이전트에게 LLM과 상호작용하는 AI 캐릭터 시스템을 탑재하고 시뮬레이션으로 관찰한 연구가 있다[2]. 연구에 따르면 25명의 에이전트가 가상 게임 공간에서 실제로 인간이 생활하는 것처럼 식사, 출근, 대화 등 행동을 한다. 벡터 데이터베이스에 관찰(행동) 내용, 초기 기억을 텍스트로 저장하고 이를 3가지 점수 (최신성, 중요성, 유사성)의 합으로 불러와 LLM으로 추론하여 다음 행동을 한다. 또한, 일정 횟수의 관측이 진행될 때마다 반영(Reflect) 기능으로 기억을 정리하고 통찰력(Insight)을 찾아 다음 행동을 계획할 때 활용한다. 이러한 기능들로 AI 캐릭터가 기억을 가지고 스스로 행동하는 시스템을 보여주었다. 그러나 이 연구는 AI 캐릭터가 인간과 상호작용하는 것이 아닌 가상의 세계에서 시뮬레이션에 초점을 두었다. AI 캐릭터가 인간과 상호작용하기 위해선 보완이 필요하다.

Ⅲ. AI 캐릭터 시스템

3.1 시스템 아키텍처

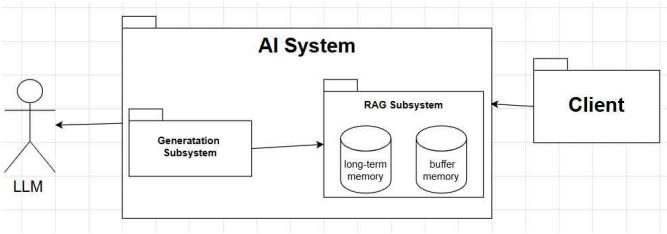

그림 1 아키텍처도

그림 1은 본 연구에서 구현한 아키텍처도이다. AI 시스템을 기억을 관리하는 RAG(Retrieval-Augmented Generation) 서브시스템과 답변을 생성하는 생성 서브시스템(Generation Subsystem) 으로 분할한다. 먼저 RAG 서브시스템은 크게 기억 저장소 생성, 기억 저장, 기억 불러오기 등 3가지 주요 기능을 수행한다. 기억 저장소 생성은 사용자 전용 AI 캐릭터의 기억을 저장할 저장소를 만드는 기능이다. [2]에서는 한 에이전트가 하나의 저장소만 가지고 기억을 저장하고 선별하여 불러오는데, 이 경우 직전 대화 내용이 빠진다면 AI 캐릭터가 흐름에 맞지 않는 대답을 한다. 본 시스템에서는 [2]에서 정의한 저장소를 장기 기억으로 두고 버퍼(buffer)라는 임시 기억 저장소를 따로 생성하여 실제 사람의 단기 기억에 해당하는 저장소를 생성한다. 모든 기억은 처음에 단기 기억에 저장되고, 답변을 생성할 때 단기 기억은 선별하지 않고 모두 불러온다. 기억 저장은 대화를 진행하면서 발생하는 상황, 대화 내용 등의 자연어를 텍스트 임베딩하여 단기 기억에 저장한다. 기억 불러오기는 AI 캐릭터가 답변을 생성하기 위하여 기억 저장소에서 기억을 불러오는 기능이다. 단기 기억에서는 모든 기억을 불러오지만 장기 기억으로부터는 어떠한 기억을 불러와야 더 자연스러운 답변을 생성할지 고민이 필요하다. 본 시스템에서는 기억이 최신 기억인지, 기억이 중요한 내용인지, 질문한 내용과 비슷한 기억인지 등 3가지 기준으로 기억 불러오기를 함으로써 더 관련된 기억을 사용하여 적절한 답변을 생성한다.

```
function prioirty_score_update(memory){
   min = Min(memory["id"])
   max = Max(memory["id"])
   memory["importance"] = queryChatGPT(memory);
   memory["recency"] = 0.7+((memory[i])["id"]-min)/(max-min)
   memory["similarity"] = chromaDBQuery(query)
   memory["prioirty"] += memory["importance"]
   memory["priority"] += memory["recency"]
   memory["priority"] += memory["similarity"]
}
```

그림 3 세 가지 점수 계산 알고리즘

그림 2는 사용한 3가지 기준의 의사 코드(Pseudo Code)이다. 중요도는 LLM 모델에게 프롬프트 엔지니어링을 통해 해당 문장이 얼마나 중요한 문장인지 질의를 하여 계산한다. [2]에서는 규칙적인 행위는 점수를 낮게 부여하고 특별한 이벤트에 높은 점수를 매겼다. 본 시스템은 대화의 맥락이나 흐름 상 중요한 기억에 높은 점수를 매겼다. 최신성은 [2]에서는 가상의 세계에 24시간 시간을 도입하여 이를 기준으로 최신 기억에 높은 점수를 매긴 반면, 본 시스템은 기억이 저장된 순서인 인덱스를 기준으로 계산한다. 0.7의 기본 값을 정한 뒤, 기억 인덱스 번호에 따라 0부터 0.3까지의 고유한 값을 계산하여 0.7 ~ 1.0 의 사이 값으로 정규화를 진행하였다. 마지막으로 유사성은 Chroma에서 제공하는 코사인 유사도를 적용하여 계산했다. 최종적으로 세 가지의 값을 더하여 기억의 우선순위를 정하였다. 생성 서브시스템은 기억 저장소에서 기억을 불러오고 프롬프트 엔지니어링을 통해 답변을 생성하여 사용자에게 보여주는 서브시스템이다. 본 시스템은 앞에서 언급한 기준을 적용하여 장기 기억으로부터 관련된 기억 30개를 가져오고, 단기 기억에 있는 정보를 모두 불러와서 답변을 생성하도록 프롬프트를 지원한다. 또한 단기 기억들을 LLM 모델을 이용해 더 고차원적인 기억으로 재구성 한 뒤 장기 기억으로 보내는 반영 과정도 구현했다. 본 시스템에서는 품질을 더욱 높이기 위하여 공감과 공통 관심사를 파악하고, 나의 경험을 공유하고, 자연스러운 질문을 하는 과정을 프롬프트 엔지니어링하였다. 이러한 요소들을 도입하여 AI 캐릭터와 대화가 더욱 자연스럽고 친밀감이 느껴지도록 하였다.

3.2 시스템 구현

본 AI 캐릭터 시스템을 직접 간단한 챗봇 서비스에 적용하여, 사용자별로 어떤 사용 경험을 얻게 되는지를 보인다. AI 캐릭터의 이름은 '연아'이며, 사용자의 이름은 '지성'으로 설정한다. 그리고 사용자1은 게임과 만화책을 좋아하는 설정으로 사건을 진행하고, 사용자2는 축구를 좋아한다는 설정으로 사건을 진행한다.

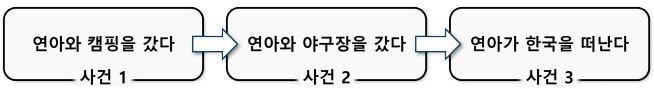

그림 3 사건 흐름도

챗봇 서비스에서 사용자는 그림 3의 3개 사건을 진행하며 연아와 대화를 주고받는다. 사건의 큰 주제가 정해져 있어 서두에는 고정된 스크립트가 출력되지만, 이후에는 사용자가 자유롭게 대화를 진행할 수 있다. 사용자는 각자의 설정에 맞게 내용을 입력하며, 입력된 대화 내용은 사용자별로 기억 시스템에 나뉘어 저장된다. 그림 4와 그림 5에서는, 사건 3의 고정된 질문에 대하여 사용자1과 사용자2 모두 같은 문장을 입력했음에도 서로 다른 답변이 생성된 것을 보여주고 있다. 이는 앞선 사건들을 진행하며 사용자별로 기억 시스템에 다른 정보가 저장되었기 때문이다. 그림 4에서는 사용자1과 함께 게임을 플레이하기로 약속했던 기억을 대화에 활용하고 있으며, 그림 5에서는 사용자2가 함께 축구를 하자고 제안했던 기억을 적절히 떠올렸다. 간단한 사례를 통해서 AI 캐릭터가 사용자 개인에 맞춤화되어 일관적이고 자연스러운 답변을 생성하는 것을 알 수 있다.

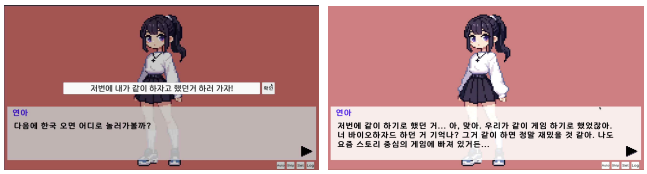

그림 4 사건 3에서의 사용자1 대화

그림 5 사건 3에서의 사용자2 대화

3.3 성능 비교

본 AI 캐릭터 시스템은 벡터 데이터베이스에서 기억을 선별하여 가져오므로 사용자와의 대화를 모두 프롬프트에 컨텍스트로 넣지 않아도 된다. 이 때문에 입력 토큰의 수를 줄일 수 있고 ChatGPT와 같이 토큰 당 요금을 부과하는 LLM 사용 비용을 절약한다. 얼마나 입력 토큰을 줄이는지 알아보기 위해 2개 사례를 두어 비교하였다. 사례 1은 사용자와 대화를 나눈 모든 대화를 프롬프트에 입력한 경우이고, 사례 2는 사용자와 대화를 파일로 만들어 ChatGPT의 자체적인 파일 검색(File search) API[3]를 이용하여 프롬프트로 만든 경우이다. 임의로 약 100개의 대화 내용과 사용자의 질의를 동일하게 주어 답변을 생성할 때 사용한 입력 토큰 양을 측정하였다.

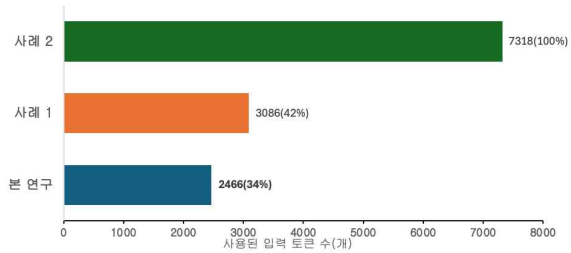

그림 6 사용된 입력 토큰 수 비교 그래프

그림 6은 사용된 입력 토큰 수를 비교하였다. 사례 2에 비하여 본 AI 캐릭터는 답변을 생성해 내는데 입력 토큰을 약 34%만 사용했고 사례 1에 비해서도 20% 적게 토큰을 사용했다.

Ⅳ. 결론

본 논문에서는 자연스러운 대화가 가능한 LLM 기반 AI 캐릭터 시스템을 소개하였다. 현재 챗 서비스의 한계를 극복하기 위해, 사용자와의 일관적인 대화를 가능하게 하는 기억 시스템을 추가하였다. AI 캐릭터 시스템은 사용자 전용 기억 저장소를 통해 단기 기억과 장기 기억을 관리하고, 최신성, 중요성, 유사성 기준으로 기억을 불러와 일관성 있는 답변을 생성한다. 또한 챗봇 서비스 사례를 통해 사용자에게 맞춤화된 경험을 제공하고, 입력 토큰 개수를 줄여 비용을 절감할 수 있다는 것을 보였다.

참 고 문 헌

[1] K. Cai, "Character.AI's $200 Million Bet That Chatbots Are The Future Of Entertainment", https://www.forbes.com/sites/kenrickcai/2023/10/11/character-ai-chatbots-group-chat/?sh=2ee23aa928f3&ref=blog.character.ai, Forbes, 2023.

[2] J. Park, J. O'Brien, C. Cai, M. Morris, P. Liang, and M. Bernstein, "Generative agents: Interactive simulacra of human behavior." Proc. of the 36th annual acm symposium on user interface software and technology. 2023.

[3] "File search", https://platform.openai.com/docs/assistants/tools/file-search, OPENAI API docs, 2024.