개인화된 AI 챗봇을 위한 지식 그래프 기반 에피소드 기억을 사용한 RAG 시스템

신지호, 이세영, 최명재, 이병정

서울시립대학교 컴퓨터과학부

sjm010529@gmail.com, ocdee39@gmail.com, paul.stark15@gmail.com, bjlee@uos.ac.kr

RAG System using Knowledge Graph Based Episode Memory for Personalized AI Chatbots

Jiho Shin, Seyeong Lee, Myoungjae Choi, Byungjeong Lee

Dept. of Computer Science and Engineering, University of Seoul

요 약

RAG는 검색 및 생성 기법으로 사전에 정리된 문서를 저장하고 LLM이 이를 활용하여 사용자에게 정확한 정보를 제공한다. 본 연구에서는 개인화된 AI 챗봇을 위한 지식 그래프 기반 에피소드 기억을 사용한 RAG 시스템 제안한다. 본 RAG 시스템은 지식 그래프 기반 에피소드 기억을 사용하여 정확하고 효율적으로 연관된 기억을 검색하여 AI 챗봇의 지속적인 대화를 지원한다. 또한 성능 평가를 위해 한국어 멀티턴(multi-turn) 대화 내용으로 검증 질문과 정답 답변을 추출하고, 검증 질문에 대해 생성한 답변이 정답 답변과 얼마나 유사한지 실험을 수행하였다. 실험을 통해 본 RAG 시스템이 ChatGPT RAG과 유사한 정확도를 유지하면서 평균 토큰 사용량은 54% 절감했다.

Ⅰ. 서 론

현재 RAG는 검색 및 생성 기법으로 사전에 정리된 문서를 저장하고 LLM이 이를 활용하여 사용자에게 정확한 정보를 제공한다. 이러한 RAG는 ChatGPT, Claude 등 정보를 얻기 위한 일회성 응답을 위주로 수행하는 챗봇과 결합하여 다양한 도메인에서 활용할 수 있다. 하지만 고객 상담 챗봇, 개인 비서 챗봇 등의 개인 맞춤화 챗봇 서비스에서는 주입된 정보와 더불어 지속적인 대화를 기억하고 활용하는 시스템이 필요하다. 이러한 배경에서 본 연구는 에피소드 기억 및 지식 그래프에 기반한 RAG 시스템을 제안한다. 에피소드 기억은 인간이 개인 경험을 부호화하고 회상하는 능력이다[1]. 최근 LLM이 마치 인간의 에피소드 기억처럼 연속된 대량의 기억을 효과적으로 저장하고 검색하는 연구가 진행되었다. 본 연구는 개인화된 AI 챗봇을 위한 RAG 시스템을 소개하고 공개된 한국어 대화 데이터로 연속된 대화에서 정확한 정보를 제공하는지 검증한다.

Ⅱ. 관련연구

EM-LLM은 연속된 대화를 토큰 단위로 새로운 맥락이 생기는 지점을 분석하여 토큰의 묶음인 에피소드를 바탕으로 기억 시스템에 저장한다[2]. 에피소드로 기억을 저장하고 검색하는 방식은 연속된 대화가 이어지는 AI 챗봇에게 효과적이다. 그러나 토큰 단위로 분할한 에피소드가 반드시 언제, 어디서, 무엇과 같은 육하원칙을 함께 기억하는 것은 아니기 때문에 토큰 단위가 아닌 자연어로서 문장의 단위로 에피소드가 저장될 필요가 있다[3]. 문장은 토큰보다 큰 단위이므로 효율적인 기억 검색을 위해선 지식 그래프를 활용하는 방법이 있다. CommunityKG-RAG는 사실 확인을 위해 여러 아티클의 문장을 각각 두 개의 노드(주어, 목적어)와 이를 연결하는 간선(서술어)로 변환하여 메모리에 저장한다[4]. 또한, 유사한 노드의 묶음인 커뮤니티를 생성하여 효율적인 지식 그래프 검색 방법을 제안했다. 그러나, CommunityKG-RAG는 사실 확인을 위해 만들어진 시스템으로 챗봇과 같이 연속된 대화를 저장하기 위해선 에피소드와 같은 개념이 결합될 필요가 있다. 본 연구는 관련연구의 장점을 통합하여 문장 단위로 에피소드를 분할하고, 지식 그래프로 기억을 검색하는 개인화된 AI 챗봇을 위한 RAG 시스템을 소개한다.

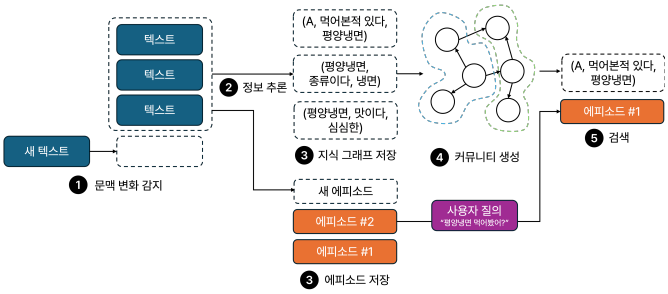

그림 1 지식 그래프 기반 에피소드 기억을 사용한 RAG 시스템 개요

Ⅲ. 지식 그래프 기반 에피소드 기억을 사용한 RAG 시스템

그림 1은 에피소드 기억 및 지식 그래프 기반 RAG를 설명하기 위한 개요 그림이다. 동작과정은 5 단계로 구성되고 이해가 쉽도록 알고리즘 1에서 그림 1의 내용을 의사 코드로 나타내었다. AI 챗봇은 사용자의 입력(q)과 아이디(userId)를 입력받아 실시간으로 대화 내용을 지식으로 업데이트하고, 최종적으로 기억을 검색하여 답변(a)을 생성한다. 단계 1, 문맥 변화 감지는 사용자의 입력(새 텍스트)과 사용자 아이디로 저장된 이전 텍스트를 비교하여 문맥이 변화했는지 부울 타입으로 반환한다(라인 1). 작업의 정확도를 높이기 위해 한국어 멀티세션 대화 데이터셋[5]을 이용해 대화가 끝난 지점을 라벨링하여 약 100개의 데이터를 만들었다.

알고리즘 1: AI 챗봇 기억시스템

Input: 사용자 입력 q, 사용자 아이디 $userId$
Output: 생성된 답변 a

```
1 flag ← CheckContextChange(q, userId);
2 if flag then
3     tuples ← ExtractRelations(userId);
4     episodeId ← CreateEpisode(userId);
5     GenerateKnowledgeGraph(userId, tuples,
        episodeId);
6     DetectCommunity(userId);
7 memories ← Retrieve(q, userId);
8 a ← GenerateResponse(q, memories);
9 return a;
```

이를 이용해 ChatGPT-4o-mini 모델을 미세조정(fine-tuning)하였다. 만약 문맥이 변화했다면 단계 2-4를 진행한다(라인 2). 단계 2, 정보 추출은 사용자의 이전 텍스트로부터 의미 있는 정보를 추출하여 지식 그래프에 저장하기 위한 튜플 형식으로 만드는 과정이다(라인 3). 추론할 수 있는 정보를 퓨샷 러닝(few-shot learning)과 프롬프트 엔지니어링을 통해 LLM이 추출한다. 각 문장은 주어, 서술어, 목적어로 세 개의 요소를 가진 튜플(트리플)로 변환한다. 단계 3, 사용자의 이전 텍스트들을 묶어 에피소드로 만들어 관계형 데이터베이스에 저장하고, 앞서 추출한 튜플로 지식 그래프를 생성한다(라인 4-5). 이때, 주어와 목적어는 각각 노드가 되고 서술어는 주어에서 목적어로 연결하는 간선이 된다. 각 노드들은 벡터로 임베딩되어 저장되고, 간선에 에피소드 ID를 프로퍼티로 저장한다. 사용자의 이전 텍스트가 저장된 임시 공간은 비운다. 단계 4, 커뮤니티 생성으로 지식 그래프에서 비슷한 노드끼리 그룹으로 묶는다(라인 6). K-Means 알고리즘을 사용하여 벡터 공간 상 인접한 노드끼리 커뮤니티가 된다. 이때, 실험적으로 커뮤니티 개수는 전체 노드의 1/3이 적합하다. 만들어진 커뮤니티는 대표값으로 각 노드의 벡터 임베딩 평균 값을 가진다. 단계 5, 검색은 사용자의 입력과 유사한 기억을 불러온다(라인 7). 사용자의 질의를 벡터로 임베딩한 후 K-Means로 가장 유사한 커뮤니티를 찾는다. 커뮤니티에 속한 모든 노드를 가져와 각 노드와 연결된 깊이 1의 노드만 다시 자연어로 바꾼다. 예를들면, “평양냉면” 노드가 찾은 커뮤니티에 속해 있다면 “평양냉면”과 연결된 “심심한” 노드와 “맛이다” 간선을 통해 “평양냉면 심심한 맛이다”라는 자연어가 검색된다. 또한, “맛이다” 간선에 프로퍼티로 저장된 에피소드 ID로 해당 에피소드를 근거 문장으로 같이 가져온다. 앞서 입력된 사용자 입력과 검색된 에피소드를 함께 프롬프트로 합친 후 LLM이 답변(a)을 생성한다(라인 8).

Ⅳ. 구현

그림 2는 본 연구에서 구현한 개인화된 AI 챗봇을 위한 RAG 시스템 아키텍처이다. Chatbot Controller는 외부 클라이언트의 입력을 받아 챗봇을 조작하며, 이 과정에서 필요한 기능을 사용하기 위해 다른 3개의 서브 시스템을 참조한다. Episode Manager는 챗봇의 대화 기록을 에피소드 단위로 관리한다. MySQL 데이터베이스를 이용하여 사용자별로 에피소드 기억과 이전 텍스트들을 각각 테이블에 저장한다. LLM Controller는 맥락 변화 검사, 에피소드에서 정보 추론 및 튜플 추출, 답변 생성 등 LLM이 필요한 작업을 수행한다. LLM은 API로 접근하며 본 연구에서는 ChatGPT를 사용한다. 그리고 Knowledge Manager는 그래프 데이터베이스인 Neo4j와 통신하여 튜플 형식의 데이터를 지식 그래프로 생성하거나 사용자의 입력과 유사한 기억 검색의 작업을 수행한다.

Ⅴ. 실험

본 연구의 RAG를 평가하기 위해 가상의 캐릭터와 사용자 간 멀티턴(multi-turn) 대화 데이터인 OPLEA[6]를 재가공하여 3761개의 턴을 가진 대화 데이터를 사용한다. 멀티턴 대화 데이터를 사전에 입력하여 2243개의 노드가 만들어졌고, 이 중 14개의 노드를 예시로 나타내었다[1]. 최근 연구[7]에서 ChatGPT로 테스트 데이터 셋을 만든 것과 유사하게 3751개

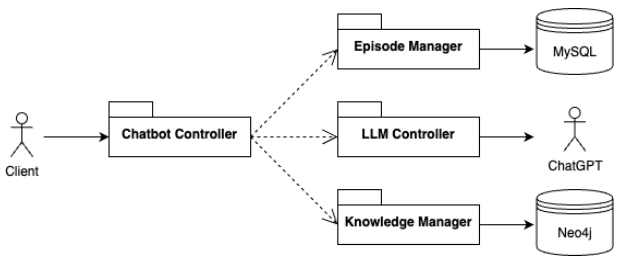

그림 2 시스템 아키텍처

표 1 시스템 성능 비교

시스템	정확도	사용한 토큰 개수 (평균)
ChatGPT	36.4	65,480
ChatGPT RAG	44.3	11,622
본 연구의 RAG	42.6	**5,293**

의 멀티턴 대화 데이터에서 43개의 검증 질문과 정답 답변을 생성한다. 또한, ChatGPT가 각 검증 질문에 RAG로 생성한 답변을 정답 답변과 얼마나 유사한지 0-100점의 범위로 평가하고 이를 평균을 내어 정확도 점수로 측정한다. 비교군으로 ChatGPT의 자체적인 RAG을 사용한 버전과 RAG를 사용하지 않고 모든 대화 내용을 프롬프트로 입력한 ChatGPT를 두었다. ChatGPT RAG는 대화 내용을 파일로 전달하면 파일의 정보를 검색하여 답변에 반영한다. 표 1은 43개의 검증 질문에 대하여 생성한 답변의 정확도 점수와 각 검증 질문에 답변하기 위해 사용된 평균 토큰 개수이다. 일반적으로 RAG에서 토큰 사용량이 크다면 불러온 정보량이 많다는 의미이기 때문에 정확도가 좋아지는 경향이 있다. 본 연구의 RAG는 ChatGPT RAG에 비해 사용한 토큰 개수는 54% 절감하였지만 정확도는 크게 차이가 나지 않았다. 43번의 테스트를 수행하는 동안 gpt-4o-2024-08-06 모델을 기준으로 RAG를 사용하지 않은 ChatGPT는 약 7달러, ChatGPT RAG는 약 1.25달러를 사용했다. 반면 본 연구의 RAG는 오직 약 0.57달러만 사용함으로써 비용도 효과적으로 절감되었다. 대화를 선별하지 않고 모두 프롬프트에 넣은 표 1의 1행의 경우, LLM의 문맥 길이(Context Length) 한계로 인해 정확도가 크게 떨어졌다.

Ⅵ. 결론

본 연구에서는 개인화된 AI 챗봇을 위한 지식 그래프 기반 에피소드 기억을 사용한 RAG 시스템을 소개하였다. 연속적인 대화가 끊임없이 이어지는 개인화 AI 챗봇에서 정확하고 효율적으로 연관된 기억을 가져오기 위하여 에피소드 기억 및 지식 그래프 기반 RAG를 도입했다. 또한, LLM과 RAG를 결합하여 하나의 프레임워크로 구현하여 사용을 용이하게 했다. 한국어 멀티턴 데이터 셋으로 검증한 결과 ChatGPT RAG에 비해 정확도 하락은 최소화되면서 토큰 사용량은 54% 절감하였다.

참 고 문 헌

[1] B. Dickerson and H. Eichenbaum, “The Episodic Memory System: Neurocircuitry and Disorders,” Neuropsychopharmacology REVIEWS, vol. 35, pp. 86-104, 2010.

[2] Z. Fountas et. al. “Human-like Episodic Memory for Infinite Context LLMs”, arXiv:2407.09450, 2024.

[3] Reviewer Sytr, “Human-like Episodic Memory for Infinite Context LLMs Official Review of Submission12669”, https://openreview.net/forum?id=BI2int5SAC, OpenReview, 2024.

[4] R. Chang and J. Zhang, “CommunityKG-RAG: Leveraging Community Structures in Knowledge Graphs for Advanced Retrieval-Augmented Generation in Fact-Checking”, arXiv:2408.08535, 2024.

[5] “한국어 멀티세션 대화”, https://www.aihub.or.kr/aihubdata /data/view.do?currMenu=115&topMenu=100&dataSetSn=71630, AI Hub, 2022.

[6] Y. Lee, W. Cho, S. Bae, H. Choi, J. Park, N. Kim and S. Hahn, ““Feels Like I’ve Known You Forever”: Empathy and Self-Awareness in Human Open-Domain Dialogs”, Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 44, 2022.

[7] B. Benoit et. al. “Generative AI to Generate Test Data Generators”, IEEE Computer Society Press, vol. 41, 2024.

1)https://github.com/IrumaeGPT/Non-Impulsive-AI/blob/main/images