RAG 기반 대학 맞춤형 LLM 챗봇 시스템 설계 및 구현

고민재, 김은별, 문민규, 방효진, 나웅수
국립공주대학교 소프트웨어학과
Email: zxc291142044@gmail.com, qwr0113@naver.com, germ47@naver.com, nijoyh0503@gmail.com, wsna@kongju.ac.kr

***Abstract*—본 논문은 대학 구성원의 학사 및 생활 정보 탐색 비용을 절감하고, 최신 정보에 근거한 응답 신뢰성을 확보하기 위해 RAG(Retrieval-Augmented Generation) 기반 LLM 챗봇 시스템을 마이크로서비스 구조로 설계 및 구현한 결과를 제시한다. 프론트엔드(React), 백엔드(Spring Boot API Gateway), AI 검색·생성 서버(FastAPI + SentenceTransformer + Qwen LLM), 벡터 데이터 저장소(Milvus)를 완전히 분리 배포하여 장애 격리, 선택적 확장, 그리고 서비스별 독립 배포가 가능한 구조를 구축하였다.**
Selenium 기반 증분 크롤링, 벡터 임베딩 저장, 의미 기반 검색, 그리고 프롬프트 기반 LLM 응답 생성을 통합한 RAG 파이프라인을 통해 정확도 높은 근거 기반 답변을 생성한다.

***Index Terms*—Retreval-augmented Generation(RAG), 대형 언어 모델(LLM), 챗봇, Microservice, 벡터 데이터베이스, 문장 임베딩(sentence embedding), 자연어 처리(Natural Language Processing)**

I. 서론

대학 홈페이지는 학사 일정, 시설 안내, 기숙사, 식단, 장학, 셔틀버스, 공지사항 등 방대한 정보를 제공하지만, 메뉴 깊이가 깊고 정보가 여러 페이지에 분산되어 있어 사용자 탐색 비용이 높다.

기존 FAQ 또는 규칙 기반 챗봇은 주로 사전 정의된 질의-응답 쌍과 패턴 매칭에 기반한 반복적인 질문에 빠르게 응답하는 데 초점을 두어 비정형 질의와 복합 맥락을 충분히 처리하기 어렵고, 범용 LLM은 내부 대학 정보의 최신성과 사실 정확성을 보장하기 어렵다 [1], [3].

이에 본 연구에서는 대학 도메인에 적합한 LLM과 벡터 검색 기반 RAG 챗봇을 마이크로서비스로 분리 설계하여, 최신 데이터에 근거한 신뢰성 높은 응답과 시스템 확장성을 동시에 확보하는 것을 목표로 한다. [2]

II. 관련 연구

RAG는 외부 검색 근거를 프롬프트에 주입하여 생성 모델의 사실 정확성과 맥락 적합성을 향상시키는 구조로, 도메인 특화 QA 시스템에서 강점을 보인다. 이러한 구조에서 고품질 의미 임베딩과 효율적인 유사도 검색은 핵심 요소이며, SBERT와 같은 문장 임베딩 기법이 널리 활용되고 있다 [4].

최근 대학 챗봇 연구는 딥러닝과 자연어 처리를 기반으로 FAQ 자동 응답과 캠퍼스 길 안내, 정보 조회를 지원하는 시스템을 제안하고 있으며 [2], [3], LLM을 활용한 고도화 연구도 점차 보고되고 있다. 하지만 대학 내부 지식 저장소와의 분리된 검색·생성 구조, 장애 격리, 독립 확장성 등을 포함한 마이크로서비스 기반 구현 사례는 여전히 제한적이다.

본 연구는 검색(Vector DB)–생성(LLM)–클라이언트–API Gateway를 완전 분리된 서비스로 구현하여, 실제 서비스 적용 가능한 구조 설계 방안을 제시한다.

III. 시스템 구조

A. 전체 아키텍처

제안 시스템은 프론트엔드, 백엔드, AI 서버, 데이터 저장소의 네 계층으로 구성된다. 프론트엔드는 사용자의 자연어 질의를 입력받아 API 형태로 백엔드에 전달하며, 백엔드는 요청을 검증한 뒤 AI 서버로 전달한다. AI 서버는 RAG 파이프라인을 수행하여 응답을 생성하고 결과를 반환한다.

- Frontend Service: React SPA, 모바일 및 웹 UX 제공
- Backend Service (API Gateway): Spring Boot, 인증·세션 검증·요청 라우팅
- AI Service: FastAPI 서버에서 RAG 검색 + Qwen LLM 응답 생성 수행
- Vector DB Service: Milvus 컬렉션 저장 및 Vector Search 전용 처리

Fig. 1. 프론트엔드 → API Gateway → AI Service → Milvus VectorDB 요청 흐름

B. 요청 흐름 분리 설계

사용자 요청은 다음 순서로 처리된다.

1) 사용자가 UI에 자연어 질문 입력
2) FE Service가 Gateway의 '/api/chat/send'로 요청 전송
3) Gateway는 세션·입력 검증 후 '/api/ai/query'로 라우팅

4) AI Service는 질문을 'SentenceTransformer'로 임베딩하여 고차원 벡터 공간으로 변환 [4]
5) Milvus에서 Top-k 유사 문단 검색 (Cosine Similarity)
6) 검색된 문맥을 LLM 프롬프트에 주입
7) Qwen LLM이 최종 답변 생성
8) Gateway → FE로 응답 반환 및 UI 출력

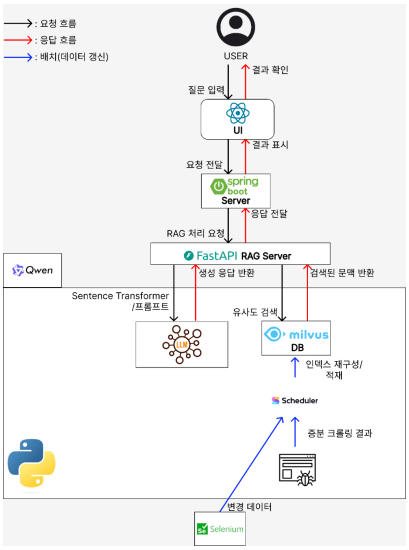

Fig. 2. 요청 흐름 시스템 다이어그램

C. 데이터 수집 및 전처리

공주대학교 공식 홈페이지를 대상으로 Selenium 및 BeautifulSoup 기반 웹 크롤러를 구현하였다. 수집된 데이터는 JSON 형식으로 저장되며, 중복 제거와 최신성 유지를 위해 주기적 크롤링과 해시 기반 비교를 수행한다. 전처리된 문서는 문단 단위로 분할되어 임베딩 대상이 된다.

IV. 구현 및 실험

선행 연구에서는 대학 FAQ 응답 [1], 캠퍼스 보조 서비스 [2], 학사 정보 관리 [3]를 위한 챗봇이 제안되었으나, 이들은 주로 단일 애플리케이션 혹은 모놀리식 구조에 기반한다. 본 연구는 프론트엔드–Gateway–AI Service–Vector DB를 완전 분리된 마이크로서비스로 구현하여, 서비스별 독립 배포와 선택적 확장성을 강조한다.

A. 구현 환경

시스템은 Ubuntu 20.04 LTS 환경에서 구축되었으며, Python 3.8과 Java 17을 기반으로 개발하였다. AI 모델 처리를 위한 FastAPI 서버와 API 게이트웨이 역할의 Spring Boot 서버로 이중 구성하여 안정성과 확장성을 확보하였다. LLM은 Qwen 계열 모델을 사용하였고, 벡터 데이터베이스는 Docker 환경에서 Milvus를 통해 운영하였다.

Operating System	Ubuntu 20.04 LTS
Development Language	Python 3.8, Java 17
AI Framework	Ollama (Qwen), SentenceTransformer
Database	Docker, Milvus Vector DB

TABLE I
SERVER ENVIRONMENT

클라이언트는 React 기반 웹 애플리케이션으로 구현되었으며, 사용자 인터페이스 제공과 실시간 챗봇 상호작용을 담당한다. 입력 장치로는 텍스트 입력창과 바로가기 버튼을, 출력 장치로는 응답 메시지와 시각적 컴포넌트를 사용하였다.

Developmetn Language	JavaScript
Framework	React, Node.js

TABLE II
CLIENT ENVIRONMENT

B. 성능 평가

프롬프트 버전(v0–v3)에 따른 응답 정확도를 비교한 결과, 구조화된 프롬프트와 RAG를 적용한 경우 정확도가 약 40%p 이상 향상되었다. 또한 양자화를 통해 메모리 사용량을 줄이면서도 실시간 응답 속도를 유지하였다.

버전	설명	정확도
v0	기본 프롬프트 없음	30%
v1	역할 제시 프롬프트	47%
v2	구조화 프롬프트(역할+제약)	59%
v3	Few-shot 포함 최적화 프롬프트	73%

TABLE III
프롬프트 고도화에 따른 응답 정확도

V. 결론

본 연구는 대학 정보 QA 시스템을 RAG + LLM 기반으로 구축하고, FE–Gateway–AI–Milvus를 완전 분리된 마이크로서비스로 설계함으로써 실서비스 적용에 적합한 구조를 확보하였다. 문단 임베딩 저장, 유사도 기반 검색, 프롬프트 기반 응답 생성 구조를 통해 대학 내부 지식 탐색과 응답 정확도 측면에서 높은 품질의 근거 기반 답변을 제공할 수 있음을 확인하였다. 향후 연구에서는 추가 데이터 소스 확장과 멀티 인텐트 검색 최적화를 통해 대학 도메인 QA 시스템의 실용성을 강화할 예정이다.

ACKNOWLEDGEMENT

이 논문은 과학기술정보통신부 및 정보통신기획평가원의 대학 ICT 연구센터육성지원사업(IITP-2026-RS-2022-00156353) 및 2026년 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업 지원을 받아 수행되었음(2024-0-00073)

REFERENCES

[1] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chatbot for university related FAQs," in *Proc. 2017 Int. Conf. on Advances in Computing, Communications and Informatics* (*ICACCI*), IEEE, 2017.
[2] B. Shivashankar, S. G. Koolagudi, and others, "Deep Learning based Campus Assistive Chatbot," in *Proc. 2021 Int. Conf. on Smart Generation Computing, Communication and Networking* (*SMARTGENCON*), IEEE, 2021.
[3] G. S. S. Vikas and others, "Information Chatbot for College Management System," in *Proc. 2021 Int. Conf. on Artificial Intelligence and Smart Systems* (*ICAIS*), IEEE, 2021.
[4] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proc. EMNLP-IJCNLP*, 2019.