

# SK네트웍스 Family AI 캠프 21회차

## LLM(초거대언어모델) 프로젝트 평가

### □ 개요

- 프로젝트 명 : LLM을 연동한 내외부 문서 기반 질의응답 시스템
- 평가 산출물 : 수집된 데이터 및 데이터 전처리 문서, 시스템 아키텍처, 개발된 소프트웨어: RAG 기반 LLM과 벡터 데이터베이스 연동 구현 코드, 테스트 계획 및 결과 보고서
- 깃허브 주소 : <https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN21-3rd-3Team>
- 평가 일시 : 2026-01-12
- 평가 대상 : 박수빈, 손현우, 우재현, 이성진, 장이선, 조남웅
- 평가 강사 : 김성환

### □ 평가 기준

산출물	평가 요소	배점	점수
수집된 데이터 및 데이터 전처리 문서	<ul style="list-style-type: none"><li>- 주제에 맞는 데이터를 충분히 수집 하였나.</li><li>- 수집한 데이터를 적절히 전처리 하였나.</li><li>- 전처리 과정과 결과를 문서에 잘 기술 하였나.</li></ul>	30	27
시스템 아키텍처	<ul style="list-style-type: none"><li>- RAG 체인 구성을 잘 설계 하였나.</li></ul>	40	37
개발된 소프트웨어: RAG 기반 LLM과 벡터 데이터베이스 연동 구현 코드	<ul style="list-style-type: none"><li>- 적절한 chunking 기법을 사용하였나.</li><li>- 응답을 위한 프롬프트를 잘 구성 하였나.</li><li>- Vector Database에 embedding한 문서들을 적절히 저장하였나.</li><li>- RAG chain 이 설계대로 잘 구성되었나.</li></ul>	20	18
테스트 계획 및 결과 보고서	<ul style="list-style-type: none"><li>- application을 평가하기 위한 적절한 평가지표를 선정 하였나.</li><li>- 평가 데이터셋을 잘 구성 하였나.</li><li>- 평가 과정과 내용이 보고서에 잘 기술하였나.</li></ul>	10	6
		100	88

## □ 평가 의견

- 수집된 데이터 및 데이터 전처리 문서
  - 수집한 데이터에 대한 분석을 상세히 진행하였다.
  - 상담 대화 데이터 외에도 심리학 이론, 상담 기법, 정서 분류 체계 등과 같은 심리 관련 이론적 자료를 추가로 수집·활용하였다면, 모델이 보다 이론적 근거를 갖춘 응답을 생성하는데 도움이 되었을 것으로 판단된다.
- 시스템 아키텍처
  - 전체적인 Chain 흐름은 비교적 명확하게 구성되어 있다..
  - 각 작업들을 모듈화 하여 구현하고 RagChain 클래스에서 각 모듈들을 조합하여 최종 Chain 모델을 정의 하였는데 LangGraph를 이용하면 상태 관리 및 흐름 제어 측면에서 보다 간결하고 명확하게 구현할 수 있었을 것으로 보인다..
- 개발된 소프트웨어: RAG 기반 LLM과 벡터 데이터베이스 연동 구현 코드
  - 사용자 질의와의 유사도를 중심으로 검색이 이루어지는 구조임을 고려하면, 임베딩 단계에서는 내담자의 질문 중심으로 벡터화를 수행하고, 해당 질문을 포함한 일정 범위의 대화 맥락을 메타데이터로 저장하는 방식을 적용하였다면 검색 정확도 측면에서 성능이 개선될 수도 있지 않을까 생각된다.
  - 다양한 돌발 상황에 대한 답변을 잘 할 수 있도록 시스템 프롬프트를 잘 구성하였다.
- 테스트 계획 및 결과 보고서
  - Retriever나 모델 선정의 기준을 잘 작성하였다.
  - 다만 모델 비교 과정에서 GPT-4o, GPT-4.1과 같은 플래그십 모델과 GPT-5-nano와 같이 상대적으로 성능이 낮은 모델을 함께 비교 대상으로 선정한 것은 비교의 공정성 측면에서 다소 아쉬운 부분으로 보인다.
  - 실제 대화내용에 대한 정성적 평가와 테스트 셋을 만들어 진행하는 정량적 RAG 성능 평가가 진행되지 않았다.

## □ 선도기업 평가 의견

- [잘한 점]  
src 디렉토리 하위에 rag, database, data 폴더를 구분하고 app 폴더를 별도로 두어 비즈니스 로직과 데이터 접근 계층을 명확히 분리하였음. 이는 각 모듈의 역할과 책임을 명확히 하여 코드 복잡도를 낮추었으며, 추후 기능 확장 및 유지보수 시 모듈 간 영향도 최소화 가능함.
- [아쉬운 점]  
app/main.py 중심의 Flask 프레임워크 구조에서 LLM API 호출이나 PDF 생성 등 I/O 바운드 작업이 동기적으로 처리되고 있음. 이는 요청이 몰릴 경우 서버 전체의 응답 지연을 유발할 수 있어, 동시 접속자가 증가하는 실제 운영 환경에서의 확장성 측면이 고려되지 않은 점이 아쉬움.
- [개선되면 좋을 점]

현재 config 폴더가 존재하나, 민감 정보나 하이퍼파라미터를 환경 변수로 분리하고 Pydantic BaseSettings 등을 활용해 관리할 필요가 있음. 이는 개발/테스트/운영 환경 간 유연한 설정 변경을 가능케 하고, 잘못된 설정으로 인한 런타임 오류를 방지하는 효과가 있음.