**TAVE 프로젝트 기획안**

|  |  |  |  |
| --- | --- | --- | --- |
| 팀장 | 김현주 | 팀 구성원 | 김현주, 김규연, 박지수, 이채원, 정보경 |
| 팀 이름 | STUBO | | |

프로젝트 소개

|  |  |
| --- | --- |
| 주제 | 수험생을 위한 RAG 기반 AI 튜터링 시스템 개발 |
| 설명 | 이 심화 프로젝트는 수험생들이 더 효율적으로 공부할 수 있도록 돕는 AI 튜터링 시스템을 개발하는 것이 목표입니다. EBS 수능특강 자료를 바탕으로, 학생이 궁금한 내용을 질문하면 관련 내용을 찾아 자연스럽게 설명해주는 AI를 구현하고자 합니다. 정확한 정보 검색과 자연스러운 응답 생성을 위해 과목별로 RAG(Retrieval-Augmented Generation) 구조를 적용해, 단순한 챗봇이 아닌 실제 학습에 도움이 되는 AI를 만드는 데 중점을 두려 합니다. 또한, 이 시스템은 모바일 앱 형태로 제공되어, 수험생들이 언제 어디서든 자유롭게 활용할 수 있도록 접근성과 편의성을 높이고자 합니다. |
| 목표 | - 수능특강 기반의 RAG 질의응답 AI 튜터 시스템 구축  - 정확한 정보 검색과 자연스러운 답변이 가능한 과목별 RAG 파이프라인 구축  - 언제 어디서든 활용 가능한 모바일 앱 형태로 구현 |
| 참고자료 | - 테디노트의 ‘랭체인LangChain 노트’ <https://wikidocs.net/book/14314>  - EBS 수능특강 자료 <https://www.ebsi.co.kr>  - LangChain 공심 홈페이지 <https://python.langchain.com/docs/introduction/> |
| 준비물 | 노트북, OpenAI API Key |
| 작업환경  (사용 툴  또는 언어) | - 사용 언어 : Python  – 검색 DB : FAISS, ChromaDB  - 응답 생성 : OpenAI GPT API  - 서버 : Flask  - 모바일 앱 : Android Studio  - 서버 배포 및 API 연결 : AWS |
| 구현할 기능  또는 결과물 | 1. 수능특강 기준 RAG 질의응답 시스템  2. 과목별 모델을 최적화하여 응답의 정확도와 답변 속도를 동시에 개선  3. Multi-Modal RAG를 활용한 이미지 속 텍스트 자동 인식 및 응답  4. Flask 기반 서버 및 API 연동 / 안드로이드 앱 구현 |

세부 계획

|  |  |  |
| --- | --- | --- |
| 역할 분담 | (NLP) RAG 기반 모델 구현 / (CV) 이미지 내 텍스트, 표, 수학 식 등 추출 | |
| 주차별 계획 | 1주차 | **[프로젝트 기획]**  - 전체 시스템 구조 설계 및 역할 분담 - 수능특강 자료 수집 - 기술 스택 및 참고 자료 조사  - 기존 RAG 사례 분석 및 구현 방식 설계 |
| 2주차 | **[데이터 전처리]**  - 수능특강 PDF 자료 전처리 - 임베딩을 위한 문서 단위 구성 및 저장 |
| 3주차 | **[RAG 초기 파이프라인 구축]**  - LangChain과 FAISS를 사용해 RAG 초기 파이프라인 개발  - 사회/생명과학 과목 데이터를 이용해 질문-검색-생성 구조 구현  - 초기 질의응답 테스트 및 응답 품질 확인  - Flask 기반 질의응답 API 구현 |
| 4주차 | **[모델 성능 개선 및 구조 확장]**  - 사회/생명과학 모델을 기반으로 다른 과목들의 모델 개발  - 과목별 파이프라인 재사용 구조 설계  - 응답 정확도 향상을 위한 검색 성능 개선  - 사진 질문 입력 기능 구현  - Multi-Modal RAG를 활용하여 이미지 내 텍스트 인식 → 검색 → 응답 연결 |
| 5주차 | **[성능 테스트 및 성능 개선]**  - 질의응답 정확도 향상을 위한 하이퍼파라미터 튜닝 (e.g., top-k, similarity threshold 조정)  - Retriever 성능 개선 (BM25, hybrid search, better chunking strategy 등 실험)  - Generator 응답 품질 비교 및 응답 포맷 통일  - 프롬프트 엔지니어링 실험 (질문 유형별 프롬프트 최적화, few-shot 예시 추가 등) |
| 6주차 | **[앱 연동 및 UI 구성]**  - Android Studio로 모바일 앱 UI 설계 및 구현  - Flask 서버와 앱 간 API 연동  - 로그인, 과목 선택, 질문 입력/응답 출력 UI 완성 |
| 7주차 | **[앱 개선 및 최종 마무리]**  - 전체 앱 통합 테스트  - 모바일 앱 최종 UI 개선  - 발표자료 및 시연 영상 제작  - 사용 설명서, 기능 정리 등 문서화 |