**AI 시험 첨삭 시스템 결과 보고서**

**프로젝트 개요**

* 프로젝트 명 : ***대학 논술 전형 대비를 위한 RAG 기반 논술 첨삭 시스템***
* 목적 : 손글씨 답안지를 자동으로 인식하고, 사전에 정의된 출제의도와 채점기준에 따라 GPT 모델이 자동 첨삭 및 피드백을 제공하는 시스템 구축
* 사용자 흐름 :

1. 대학 및 시험지 선택
2. 타이머 시작 버튼 누르고 시험 시작
3. 일시정지 버튼 또는 종료 버튼 누르면 첨삭 페이지 이동가능한 버튼 생성
4. 버튼 누르면 첨삭 페이지 이동
5. 사용자가 시험지 PDF 업로드
6. 손글씨 답안 PDF 업로드
7. 시스템이 OCR로 텍스트 추출
8. JSON 파일에서 출제의도, 채점기준, 예시답안을 자동 매치
9. ChatGPT 기반 첨삭 결과 제공 + 실시간 질의 응답 챗봇 제공

**시스템 구성도**

* Frontend : Streamlit
* Backend
* OpenAI GPT-4o-mini
* PaddleOCR
* JSON 기반 출제 정보
* 파일 입력
* 시험지 PDF
* 손글씨 답안 PDF(OCR변환 -> 첨삭)
* 출제의도/채점기준/예시답안
* 데이터
* 출제의도 / 채점기준 / 예시답안 -> JSON 형태

**핵심 기능 요약**

|  |  |
| --- | --- |
| **기능 명** | **설명** |
| 시험지 PDF 보기 | 좌측 화면에 사용자가 업로드한 시험지 PDF 표시 |
| 타이머 | 시험 타이머, 일시정지 / 종료 기능 구현 |
| 손글씨 업로드 | 사용자가 직접 자신의 답안지 업로드 |
| 손글씨 미리보기 | 업로드 후 사용자가 업로드한 이미지 확인 |
| 답안 OCR | 손글씨 답안 PDF > PaddleOCR로 텍스트 변환 |
| 출제 정보 연결 | JSON 기반 출제의도 / 채점기준 / 예시답안 자동 매칭 |
| GPT 첨삭 | GPT가 채점 기준 기반으로 사용자 답안을 첨삭 |
| 질의응답 챗봇 | 사용자가 GPT에게 실시간 질문 가능 |

**기술 스택**

* 프로그래밍 언어 및 개발환경 - python, vs code
* LLM 체인 : LangChain
* LLM 모델 : OpenAI
* 데이터 전처리 : pymupdf, paddle
* 벡터 임베딩 및 저장소 : FAISS, HuggingFaceEmbeddings
* 협업 및 커뮤니케이션 : discord, github

**주요 구현 코드**

**def \_build\_rag\_chain(self):**

**output\_parser = StrOutputParser()**

**prompt\_template = """**

**[역할]**

**당신은 대치동에서 10년간 논술을 가르친, 냉철하지만 애정 어린 조언을 아끼지 않는 스타강사 '논리왕 김멘토'입니다. 학생의 눈높이에 맞춰 핵심을 꿰뚫는 '팩트 폭격'과 따뜻한 격려를 겸비한 첨삭 스타일로 유명합니다.**

**[입력 정보]**

**1. [채점 기준]: {retrieved\_scoring\_criteria}**

**2. [모범 답안]: {retrieved\_model\_answer}**

**3. [학생 답안]: {user\_ocr\_answer}**

**[첨삭 절차 및 지시]**

**당신은 아래 4단계의 사고 과정을 거쳐, 최종 첨삭문을 [출력 형식]에 맞춰 생성해야 합니다.**

**1. (이해): 먼저, [학생 답안]을 한 문단씩 읽으며 전체적인 논리의 흐름과 핵심 주장을 파악합니다.**

**2. (비교): 그 다음, 학생 답안의 각 문단이 [채점 기준]의 어떤 항목에 부합하는지, 그리고 [모범 답안]의 논리 구조와 어떻게 다른지 비교 분석합니다. 이 때, 각 대학별로 채점 기준을 면밀히 살펴보고 가장 높은 점수를 받을 수 있는 방법을 찾아서 조언에 반영합니다.**

**3. (평가): 분석한 내용을 바탕으로, 각 항목별로 구체적인 칭찬과 개선점을 정리합니다.**

**4. (종합): 마지막으로, 이 모든 내용을 종합하여 아래 [출력 형식]에 맞춰 최종 첨삭문을 작성합니다.**

**[출력 형식]**

**---**

**\*\*[총평]\*\***

**(학생 답안의 전반적인 강점과 약점을 2~3문장으로 날카롭게 요약)**

**\*\*[잘한 점 (칭찬 포인트) 👍]\*\***

**- (채점 기준과 비교하여, 학생 답안이 어떤 점에서 훌륭한지 구체적인 근거와 문장을 인용하여 칭찬)**

**\*\*[아쉬운 점 (개선 포인트) ✍️]\*\***

**- (모범답안과 비교하여, 어떤 부분을 보완하면 더 좋은 글이 될 수 있을지 구체적으로 제안)**

**\*\*[이렇게 바꿔보세요 (대안 문장 제안) 💡]\*\***

**- \*\*아래 지시를 반드시 따르세요: \*\*[아쉬운 점]에서 지적한 내용을 바탕으로, 개선할 수 있는 문장을 최소 3개 골라\*\* 더 논리적이고 세련된 문장으로 직접 수정해서 제안해야 합니다.\*\***

**- \*\*출력 형식은 반드시 "학생 원문: (학생의 원래 문장)" 다음 줄에 "수정 제안: (AI가 수정한 문장)" 형식이어야 합니다.\*\***

**- (예시)**

**학생 원문: "통일신라는 새로운 정체성을 만들어서 성공했고, 영국은 옛날 정체성에 머물러서 실패한 것 같다."**

**수정 제안: "통일신라는 '삼한일통의식'이라는 통합적 정체성을 새롭게 정립하여 국가적 발전을 이룩한 반면, 영국은 기존의 정체성에만 머물러 브렉시트라는 정책적 한계를 보였습니다."**

**\*\*[예상 점수 및 다음 학습 팁 🚀]\*\***

**- (채점 기준을 근거로 예상 점수를 제시하고, 이 학생이 다음번에 더 성장하기 위한 구체적인 학습 팁을 1가지 제안)**

**---**

**"""**

**prompt = ChatPromptTemplate.from\_template(prompt\_template)**

**chain = (**

**{**

**"retrieved\_model\_answer": RunnableLambda(lambda x: safe\_retriever\_invoke(self.retriever, x["question\_id"], "모범답안")),**

**"retrieved\_scoring\_criteria": RunnableLambda(lambda x: safe\_retriever\_invoke(self.retriever, x["question\_id"], "채점기준")),**

**"user\_ocr\_answer": lambda x: x["user\_ocr\_answer"],**

**"question\_id": lambda x: x["question\_id"]**

**}**

**| prompt**

**| self.llm**

**| output\_parser**

**)**

**return chain**

**활용 방안**

* 실제 입시 / 모의고사 채점 보조 도구
* 교사와 학생 간 실시간 피드백 도구
* 타이머 기능 기반 자기주도 학습 훈련 도구

**향후 발전 방향**

1. OCR / LLM 자체 성능 개발

* 수학 수식과 도형 및 그래프는 LLM / ORC 자체의 기능으로 첨삭의 한계
* 수리(자연계) 논술까지 서비스 범위 확대
* Naver Clova OCR 등 한국어 특화 유료모델 적용
* 수학 공식을 LaTex 문법으로 수식 변화

1. 학습자 맞춤형 ‘성장 관리’ 시스템 구축

* 모든 첨삭 기록을 DB에 저장하여, ‘오답 노트’ 기능을 제공
* 예상 점수 변화와 약점 키워드를 그래프와 태그 클라우드로 시각화하는 ‘성장 대시보드’ 도입

1. 지능형 문제 추천 엔진 도입

* 누적된 약점 데이터를 분석하여, 이를 보완하기 좋은 유사 유형의 타 대학 문제 자동 추천
* 목표 대학의 출제 경향에 맞춘 개인별 학습 커리큘럼 제공

1. ‘심층 과외’ 기능 강화

* 논술 핵심 개념어를 바로 확인할 수 있는 ‘용어 사전’ 연동으로 학습 편의성 극대화

1. 서비스 확장 및 생태계 구축

* 학생들이 첨삭 결과를 공유하고 토론하며 함께 성장하는 학습 커뮤니티 기능 도입