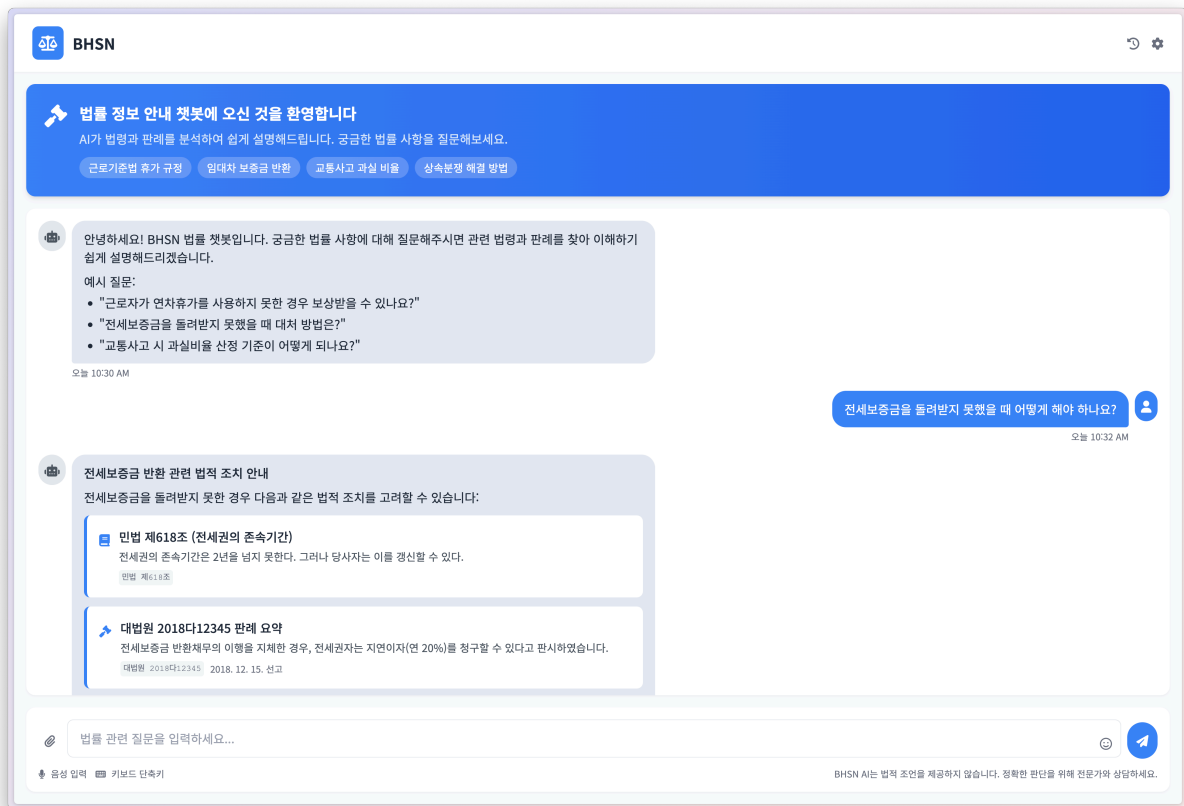


AI Model Solution Engineer 과제

개요



※ 참고용 화면으로, 실제 개발 시 동일하게 구현할 필요는 없습니다.

법률 정보의 양이 방대해지고 내용이 복잡해짐에 따라, 일반 대중이 필요한 법령·판례 정보를 찾아 이해하는 데 큰 어려움을 겪고 있습니다. 특히 판례와 같은 전문 문서는 해석이 까다로워 개인이 시의적절한 결정을 내리기 어렵습니다.

이러한 정보 비대칭 문제를 해소하고, 누구나 법령과 판례를 쉽고 빠르게 탐색·이해할 수 있도록 돕고자 저희는 AI 기술을 활용한 '법률 정보 안내' 챗봇 서비스를 기획했습니다. 이 서비스는 **법원 판례 데이터**를 기반으로 사용자의 질문과 연관된 법령·판례를 검색·요약해 실시간으로 제공함으로써 법적 쟁점을 이해하는 데 도움을 주는 것을 목표로 합니다.

본 과제에서는 '**법률 질의응답 데이터셋**'을 활용하여 **RAG(Retrieval-Augmented Generation)** 기술 기반의 법률 정보 챗봇을 구현합니다.

과제

Task 1: 법률 문서 임베딩 및 검색 인덱스 구축

- 제공되는 `cases.json` 파일에는 100개의 판례 데이터가 담겨 있습니다. 이 원시 데이터를 다음과 같은 파이프라인으로 처리해주세요.
 - JSON 데이터 파싱 및 구조 분석
 - 법률 Q&A에 활용 가능한 핵심 정보 추출 및 전처리
 - 텍스트 임베딩 모델을 통한 벡터 변환
 - 벡터 데이터베이스(Pinecone 등)에 인덱싱

과제 제출물

- `cases.json` 을 로드 → 임베딩 → Vector DB 저장 과정 전반의 **구현 코드**

Task 2: 검색 증강 생성(RAG) 기반 법률 Q&A 인터페이스 개발

- Task 1에서 구축한 벡터 인덱스를 기반으로 사용자 질문과 의미상으로 연관성이 높은 법률 정보를 검색하고, 이를 활용하여 대화형 법률 Q&A 서비스를 구현해주세요.
- 완전한 웹 애플리케이션 개발보다는 **Python 기반의 UI 프레임워크 (Streamlit, Gradio, Chainlit)** 등의 프레임워크를 활용한 프로토타입 형태로 제작하되, 핵심 RAG 로직과 데이터 플로우가 명확히 구현되어야 합니다.

과제 제출물

- RAG 기반 챗봇의 **구현 코드** 및 사용자 질문에 대한 **예시 답변 시연** (스크린샷, 콘솔 로그, 또는 챗봇 구조에 대한 간략한 명세서)

Task 3: 응답 품질 최적화 및 성능 향상

- 개발된 법률 Q&A 시스템의 답변 정확성과 신뢰성을 높이기 위한 고도화 작업을 수행해주세요. 다음과 같은 접근 방법들을 자유롭게 조합하여 활용할 수 있습니다.
 - 프롬프트 최적화
 - 데이터 청크(Chunk) 분할 및 적재 방식 최적화
 - 검색 알고리즘 개선(예: Hybrid Search)
 - Reranking 모델을 통한 검색 결과 정제
 - 답변 검증 및 할루시네이션 방지 메커니즘
- 제공되는 `qna_sample.json` 의 5개 샘플 질문을 참고하여 시스템을 테스트하고 개선해주세요.
- `qna_submission.json` 파일을 확인하시면 각 질문에 대응되는 답변이 빈 칸으로 남아있는 걸 보실 수 있습니다. 여기에 챗봇 시스템이 답변한 답변 값을 기입하여, **완성된 `qna_submission.json` 파일을 제출**해 주시면 됩니다.

과제 제출물

- 챗봇 성능 개선 방법과 적용 결과를 분석한 **자유 양식의 문서 혹은 평가 보고서**
- 평가 방법론 및 정량적 지표 제시
- 답변과 참조 판례를 채운 `qna_submission.json`

기타

- 본 과제는 여러 모델 및 API 사용이 필요합니다.
 - LLM
 - Google AI Studio 등을 통해 무료 할당량을 사용하는 것을 권장합니다.
 - 다양한 무상 API 목록은 [해당 Github Repository](#)를 참고해주세요
 - 벡터 데이터베이스
 - [Pinecone의 Starter 플랜](#)은 소규모 실험에 충분한 무료 용량을 제공합니다

- 자체 호스팅을 선호한다면 Chroma, Milvus, FAISS 등 오픈소스 대안도 고려할 수 있습니다
- Embedding API
 - 무료 서비스를 사용하거나 (e.g. Google Gemini Embedding)
 - 로컬에서 직접 호스팅하여 사용할 수 있습니다. (e.g. **text-embedding-inference** 등)
- 본 과제의 모든 자료는 외부 유출 및 제3자 제공을 금합니다.

FAQ

Q1. 판례 외에 법령 데이터도 사용해야 하나요?

A. 본 과제는 제공된 **판례 데이터(cases.json)**를 중심으로 구현합니다. 별도 법령 데이터는 과제 필수 사항이 아닙니다.

Q2. 웹 애플리케이션을 완전히 구현해야 하나요?

A. 아닙니다.. Streamlit, Gradio, Chainlit 등 Python UI 프레임워크를 활용한 프로토타입(데모) 수준이면 충분합니다.

Q3. **qna_sample.json** , **qna_submission.json** 은 어떻게 활용하나요?

A.

- **qna_sample.json**: 개발 과정에서 답변 품질 테스트 및 시스템 개선에 활용
- **qna_submission.json**: 최종 평가에 사용될 질의로, 실제 답변 결과 제출이 요구됨

Q4. 과제는 어떻게 제출하면 되나요?

A. 과제는 **담당자에게 이메일로 제출**해 주시기 바랍니다.

이메일 제출 시, 다음 사항을 지켜주세요.

- 제출 이메일 주소: **recruit@bhsn.ai**
- 제목 예시: **[BHSN][AI Model Solution Engineer 과제] 홍길동**
- 첨부 파일: 구현 코드와 성능 평가 보고서 등 과제 요구 제출물을 함께 **압축(zip)**하여 첨부