결과보고서 자료 - 유빈

결과, 방법론, 앞으로의 계획

현재 결과

- 프로젝트 목적

  의료기기 임상시험 진행 시 필요한 법적 기준과 절차가 있습니다. 이를 지키기 위해 식약처 홈페이지의 민원인 안내서 문서를 참고해서 진행합니다. 따라서 빠른 시간내에, 효율적으로 문서를 파악하고 필요한 부분을 얻기 위해서 민원인 안내서 문서 기반으로 답변하는 생성형AI를 만들기로 하였습니다.

- 프로젝트 목표

  프로젝트의 목적에 맞게 민원인 안내서 문서 기반으로 답변하는 AI를 만들어야함으로 RAG 기술을 통한 문서기반 생성형 AI를 개발.

- 초기 성과
  - 데이터 수집

    현재 식품의약품안전처 홈페이지의 법령/자료의 민원인안내서 의료기기 카테고리에 올라와 있는 문서를 직접 다운로드 하여 수집.

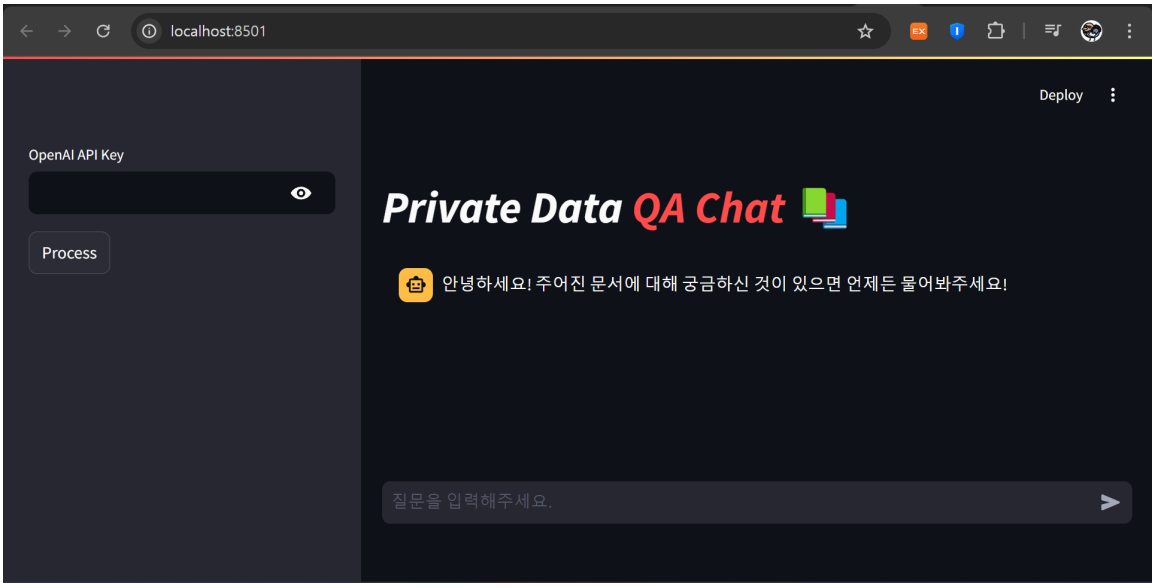
  - 비즈니스 로직 구현

    1. **사용자 인증 및 데이터 준비**
       - 사용자로부터 **OpenAI API Key**를 입력받아 대화형 AI 시스템에 필요한 키를 인증합니다.
       - 사용자가 입력한 API 키가 없으면 시스템은 작동하지 않습니다.

    2. **문서 업로드 및 처리**
       - 시스템이 처음 실행될 때, 로컬 디렉토리에 저장된 PDF, DOCX, PPTX 형식의 파일들을 검색하여 문서 내용을 불러옵니다.
       - 문서 내용은 적절한 크기로 쪼개어(900 토큰 단위) 저장되며, 이를 통해 이후 검색 및 질문 응답 시 효율적으로 사용될 수 있도록 합니다.

3. **벡터스토어(Vector Store) 생성 및 관리**
   - 문서 내용은 HuggingFace의 사전 학습된 한국어 임베딩 모델을 사용하여 벡터화한 후 **FAISS**를 통해 저장합니다.
   - 생성된 벡터스토어는 파일로 저장되며, 이후 생성된 벡터스토어가 있을 경우 해당 파일을 로드하여 사용됩니다.
4. **대화형 체인 생성 및 유지**
   - 벡터스토어를 이용해 문서 내용을 검색할 수 있는 **ConversationalRetrievalChain**을 생성합니다.
   - 대화 기록을 **ConversationBufferMemory**를 이용해 메모리에 저장하고, 대화가 진행될수록 이전 기록을 기반으로 응답을 제공합니다.
5. **질문에 대한 응답 생성 및 문서 출처 제공**
   - 사용자가 질문을 입력하면 시스템은 미리 준비된 벡터스토어를 이용해 관련 문서를 검색하고, 질문에 대한 답변을 생성합니다.
   - 답변뿐만 아니라 참고 문서의 출처도 함께 제공하며, 여러 문서에서 검색된 내용을 보여줍니다.
6. **대화 상태 관리**
   - 대화 중 사용자의 질문과 AI의 응답을 `st.session_state`에 저장하여, 세션이 유지되는 동안 이전 대화 기록을 참조할 수 있게 합니다.

- 시스템 데모



스크린샷 파일(pdf파일) 참고

방법론

- 기술적 접근
  - 모델 선택 이유

    현재 LLM 중 가장 성능이 좋고 많은 데이터로 훈련 되어있으며 좋은 퀄리티의 답변을 받을 수 있는 OpenAI의 ChatGPT 모델을 선택했습니다. 또한 API 사용으로 손쉽게 모델을 사용할 수 있었습니다. 따라서 현재 가장 성능이 좋은 GPT-4 모델을 이 프로젝트에 적용 중입니다.

  - 아키텍처 소개
    - 기술스택

      RAG를 구현하기 위해 Langchain 프레임워크를 기반으로 개발하였습니다. Streamlit 프레임워크를 기반으로 간단한 웹 인터페이스를 개발하였습니다.
    - 아키텍처 구조도 (ppt올릴 시에는 화질 좋게 svg 파일 이용)

  - 데이터 처리

    식품의약품안전처의 홈페이지의 법령/자료 카테고리에서 법령정보 > 공무원지침서/민원인안내서 중 의료기기 카테고리에 해당하는 문서를 모두 직접 다운로드 하여 수집하였습니다.

    수집한 문서를 바탕으로 Text Split > Embedding을 거쳐 벡터스토어 파일을 생성하여 Process Command 실행시 미리 만들어진 벡터스토어파일을 로드하여 사용할 수 있도록 하였습니다. 따라서 Process Command 실행 후 LLM체인 생성까지 매번 임베딩을 해야하는 문제와 로딩속도를 개선하였습니다.

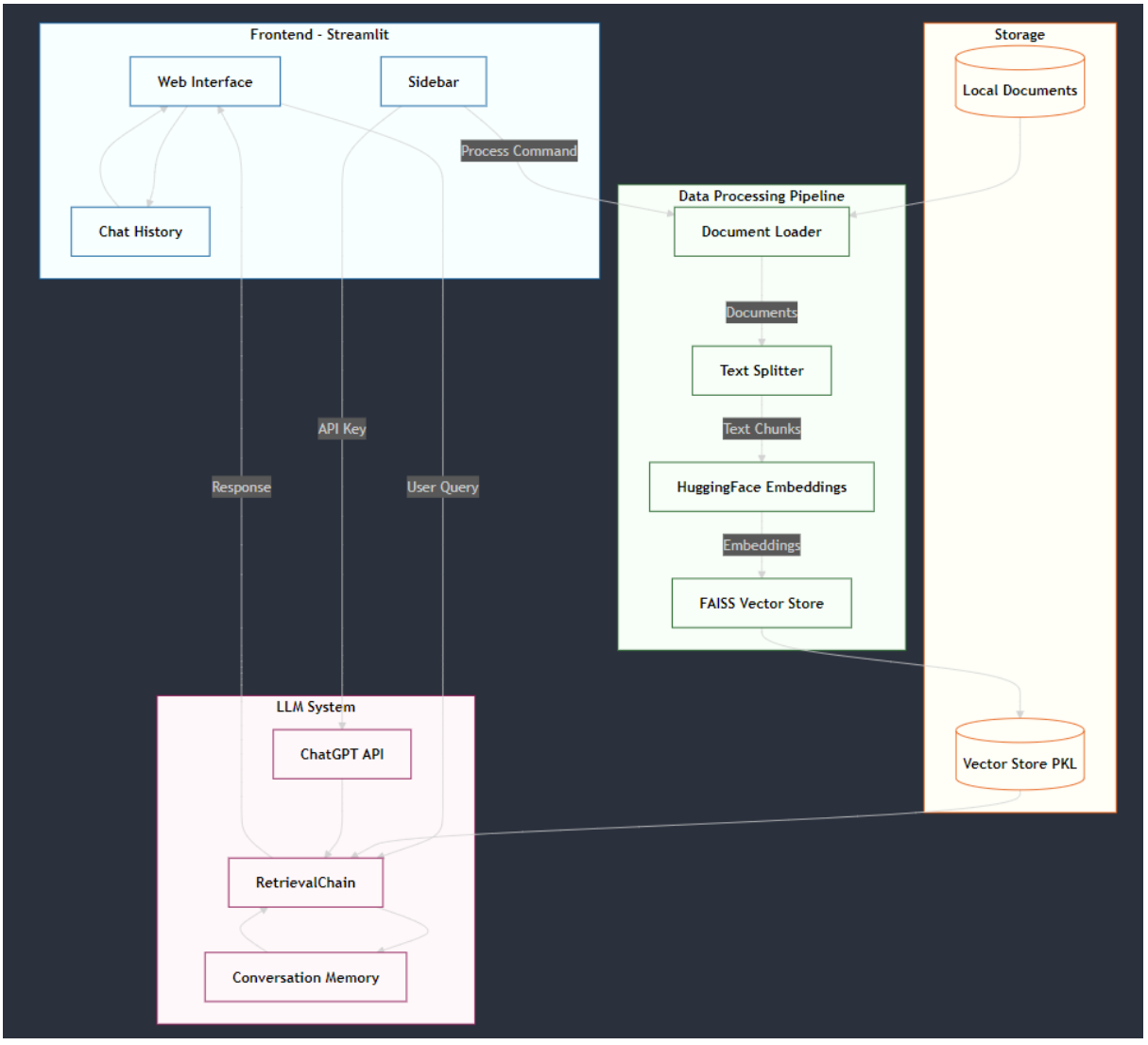
  - 모델 훈련과 검증

    현재 사용자의 질문이 들어오면 사용자의 질문 텍스트와 가장 벡터 유사도가 비슷한 텍스트 청크를 채택하여 LLM 쿼리에 추가 정보로 덧붙여 LLM에게 질문하는 방식입니다.

    시스템 데모를 보면 사용자의 질문에 맞는 문서를 채택하여 어느정도 그 문서 기반의 답변을 하고 있습니다. 그러나 사용자가 원하는 문서 기반의 흐름으로 대화를 이

끌어 가려할 때 프롬프트에 정확히 문서의 이름을 명시해 주지 않으면 단지 질문의 벡터유사도에 기반하기 때문에 상관없는 문서의 청크를 채택하는 문제점이 발생하였습니다.

mermaid-diagram-2024-10-21-230858.svg



앞으로의 계획

- 기능 확장

민원인 안내서 문서를 실시간으로 가져오도록 수정할 예정입니다.

임베딩 된 벡터스토어 파일의 크기가 너무 큰 관계로 배포시에 문제가 있었습니다. 또한 사용자가 원하는 문서를 타겟팅해서 RAG를 하고 싶은 경향이 있으므로 LLM체인 생성 이전에 사용자가 원하는 문서를 선택후 임베딩 한 다음 문서를 참고하도록 로직을 변경할 예정입니다.

- 모델 개선

  답변 퀄리티를 높이기 위해 Multi-Query Retriever를 적용할 예정입니다. 사용자의 질문을 여러개의 유사질문으로 재생성하여 벡터스토어에서 조금 더 다양하고 풍부한 청크를 채택할 수 있도록 할 예정입니다.

  또한 키워드를 고려하고 맥락을 잘 파악 하기 위해 BM25 키워드 검색 + FAISS 맥락 검색이 합쳐진 Ensemble Retriever를 적용 후 문서에서 좋은 품질의 청크를 채택하도록 할 예정입니다.