**시스템 아키텍처**

텍스트, 스크린샷, 폰트, 디스플레이이(가) 표시된 사진

자동 생성된 설명

- 주요함수

rag\_utils.py

is\_relevant(query)

주어진 쿼리(문자열)가 특정 키워드 목록 중 하나라도 포함하고 있는지를 확인하여 해당 쿼리가 관련성이 있는지를 판단한다.

respond\_to\_query(query)

사용자의 질문(query)가 응급 상황과 관련있는지 판단하여 관련있다면 기존 RAG chain이 처리할 수 있도록 None을 반환, 그렇지 않다면 응답하지 않도록 한다.

rag\_chain(query)

질문이 응급처치와 관련이 없는 경우, 미리 지정해둔 응급 상황과 관련된 질문을 해달라는 답변을 반환한다. 질문이 응급처치와 관련 있는 경우, 관련 문서를 가져와서 Rag-Chain을 통해 응답한다.

stt.py

voice\_to\_text()

녹음된 사용자 음성파일을 열어 음성인식 모델(whisper)를 사용하여 텍스트로 전환한 뒤 반환한다.

record\_audio()

이 함수는 함수는 사용자의 목소리를 녹음한 후, 해당 음성을 텍스트로 바꾸어 반환한다. 이 함수는 음성 녹음과 오디오 파일 생성, 그리고 녹음된 음성을 텍스트로 변환하는 작업을 수행한다.

st\_utils.py

print\_msg(prompt)

사용자 메시지와 그 메시지에 해당하는 AI의 응답을 출력하는 함수이다. 챗봇 특성에 걸맞게 Stream 형식으로 출력이 되도록 하였다.

streaming\_md()

웹의 main에서 텍스트 답변을 위한 AI 응답 출력 함수이다.

app.py

streamlit.chat\_input() 함수를 사용하여 챗봇 환경을 구축하였다. 사용자가 입력한 내용이 rag\_chain을 거쳐 적합한 응답을 출력한다. 사용자와 AI 아이콘에 구분을 두어 가시성을 높였다.

side bar

웹 좌측에 음성 챗봇을 구현하기 위해 녹음 버튼과 안내 문장을 출력하였다. REC 버튼을 누르면 5초간 녹음이 시작되며 만들어진 녹음 파일은 텍스트로 변환되어 rag\_chain을 통해 main 챗봇 부로 응답을 출력한다.