llm 알고리즘 문서

챗봇

1. 전체 개요

이 코드는 FastAPI를 기반으로 한 챗봇 API를 구현합니다. 사용자가 질문을 보내면 Azure AI Search와 Azure OpenAI를 연동하여 관련 정보를 검색하고, 검색된 문서를 기반으로 답변을 생성합니다.
주요 기능은 다음과 같습니다.

- **환경 변수 로드:** `.env` 파일을 이용해 필요한 API 엔드포인트, 키, 배포명 등의 설정 값을 불러옵니다.
- **Azure OpenAI 연동:** OpenAI API를 Azure 환경에서 사용하도록 설정하고, 질문에 대해 LLM을 활용해 키워드를 추출하거나 답변을 생성합니다.
- **Azure AI Search 연동:** 추출된 키워드를 기반으로 AI Search에서 관련 정보를 검색합니다.
- **답변 생성:** 검색된 데이터만을 활용하여 사용자의 질문에 대해 정책 및 사업 추천과 관련된 답변을 생성합니다.

---

2. 초기 설정

2.1. 라이브러리 임포트 및 FastAPI 애플리케이션 생성

- **FastAPI:** 웹 프레임워크로 API 엔드포인트를 생성합니다.
- **dotenv:** `.env` 파일에서 환경 변수를 로드하여 보안 및 설정 관리를 용이하게 합니다.
- **os, openai, azure 관련 라이브러리:** Azure OpenAI와 AI Search 서비스를 사용하기 위한 라이브러리입니다.

2.2. 환경 변수 로드 및 API 설정

- **Azure OpenAI 설정:**
  - `openai.api_type`을 `"azure"`로 설정하여 Azure 환경에서 OpenAI를 사용하도록 지정합니다.
  - `openai.api_base`, `openai.api_key`, `openai.api_version`을 환경 변수에서 불러옵니다.
  - 배포 이름(`azure_openai_deployment_name`)도 환경 변수로 설정하여 OpenAI 모델 호출 시 사용합니다.
- **Azure AI Search 설정:**
  - `ai_search_api_key`, `ai_search_endpoint`, `ai_search_index` 값을 환경 변수로 불러옵니다.
  - `SearchClient` 객체를 생성하여 Azure AI Search 인덱스에 접근할 수 있도록 구성합니다.

---

3. 주요 기능별 함수 설명

3.1. 키워드 추출 함수 (`extract_keywords`)

- **목적:**
  사용자의 질문에서 가장 관련성이 높은 키워드를 추출합니다.
- **동작:**
  - OpenAI의 ChatCompletion API를 사용해, 시스템 메시지와 함께 사용자 질문을 전달합니다.
  - LLM이 응답한 메시지에서 키워드를 추출해 반환합니다.
  - 예외 발생 시 에러 메시지를 반환하도록 구성되어 있습니다.

3.2. AI Search 검색 함수 (`search_in_ai_search`)

- **목적:**
  추출된 키워드를 기반으로 Azure AI Search에서 관련 콘텐츠를 검색합니다.
- **동작:**

- 먼저 `extract_keywords` 함수를 호출하여 질문에서 키워드를 뽑아냅니다.
- 추출된 키워드를 검색 쿼리로 사용하여 `search_client.search`를 호출합니다.
- 검색 결과에서 `'content'` 필드를 가진 문서를 추출해 하나의 문자열로 합칩니다.
- 관련 문서가 없을 경우 "No relevant information found." 메시지를 반환합니다.
- 예외 상황에서는 에러 메시지를 반환합니다.

3.3. OpenAI 답변 생성 함수 (`get_answer_from_openai`)

- **목적:**
  AI Search에서 가져온 데이터(문맥)를 바탕으로 사용자의 질문에 대한 답변을 생성합니다.
- **동작:**
  - 생성할 답변의 조건과 형식을 상세히 기술한 프롬프트를 구성합니다.
    - 데이터는 오직 AI Search 결과만을 기반으로 사용합니다.
    - 추가적인 외부 정보나 추측을 포함하지 않으며, "중소기업" 관련 내용은 답변에서 제외하도록 명시합니다.
    - 사람이 전달하는 것처럼 부드럽고 이해하기 쉬운 설명 형식을 갖춥니다.
  - OpenAI ChatCompletion API를 호출해 답변을 생성하고, 응답 메시지의 내용을 반환합니다.
  - 예외 발생 시 에러 메시지를 반환합니다.

3.4. 챗봇 응답 엔드포인트 (`/ask`)

- **FastAPI POST 엔드포인트:**
  사용자가 `/ask` 엔드포인트에 질문을 전달하면 아래 과정이 실행됩니다.
- **동작:**
  - `search_in_ai_search` 함수를 호출하여 질문에 대한 관련 문서를 검색합니다.
  - 만약 검색 결과가 없으면 추가 정보를 요청하는 메시지를 반환합니다.
  - 관련 문서가 있으면, `get_answer_from_openai` 함수를 호출해 최종 답변을 생성하여 반환합니다.

리포트 생성

1. 전체 개요

이 코드는 GPT 모델을 활용하여 특정 기간 동안의 데이터를 바탕으로 보고서를 생성하고, 이를 HTML로 저장한 후 PDF 파일로 변환하여 서버에 업로드하는 파이프라인을 구현합니다.
주요 프로세스는 다음과 같습니다.

- **데이터 준비:** 서버 API를 호출하여 보고서 생성을 위한 데이터를 가져옵니다.
- **GPT 응답 생성:** 전달받은 데이터와 사용자가 지정한 조건(페르소나, 기간 등)을 기반으로 GPT를 호출해 보고서 내용을 생성합니다.
- **HTML 및 PDF 변환:** GPT가 반환한 응답 중 HTML 콘텐츠를 추출하여 HTML 파일로 저장하고, 이를 PDF 파일로 변환합니다.
- **서버 전송:** 생성된 PDF 파일과 관련 메타데이터(멤버 ID, CCTV ID, 보고서 제목 등)를 서버에 POST 방식으로 전송합니다.

---

2. 주요 라이브러리 및 설정

- **pandas:** 데이터 처리를 위한 라이브러리. (현재는 주석 처리되어 있으나 추후 데이터 파일 경로와 함께 활용될 수 있음)
- **gpt_response:** GPT 모델을 호출하여 응답을 생성하는 사용자 정의 모듈.
- **pdfkit:** HTML 파일을 PDF로 변환하기 위한 라이브러리.
  - wkhtmltopdf 실행 파일 경로가 필요하며, 옵션 설정을 통해 페이지 크기, 여백, 인코딩 등을 지정함.
- **requests:** HTTP GET/POST 요청을 통해 데이터를 가져오거나 서버에 파일을 전송.

- **정규표현식(re):** GPT 응답에서 HTML 콘텐츠만 추출하기 위한 용도.
- **sys 및 os:** 모듈 경로 조정 및 파일 경로 처리.

---

3. 주요 함수 설명

3.1. `extract_html_content(response)`

- **목적:**
  GPT 응답에서 실제 HTML 콘텐츠 부분만을 추출합니다.
- **동작:**
  - 정규표현식을 사용해 `<!DOCTYPE html>`로 시작하고 `</html>`로 끝나는 부분을 찾습니다.
  - 일치하는 경우 해당 HTML 콘텐츠를 반환하고, 없으면 전체 응답을 그대로 반환합니다.

3.2. `save_html(response, html_file)`

- **목적:**
  추출된 HTML 콘텐츠를 지정된 파일 경로에 저장합니다.
- **동작:**
  - `extract_html_content` 함수를 통해 HTML 콘텐츠를 추출한 후, 지정한 파일 이름(`html_file`)으로 저장합니다.
  - 저장 후 성공 메시지를 출력합니다.

3.3. `convert_local_image_paths(html)`

- **목적:**
  보고서 내 포함된 로컬 이미지 경로를 파일 URL 형식으로 변환하여 PDF 변환 시 올바르게 표시되도록 합니다.
- **동작:**
  - 현재 작업 디렉터리의 절대 경로를 가져온 후, Windows 파일 경로를 URL 형식(`file:///...`)으로 변환합니다.
  - HTML 내 이미지 경로(`src="`)를 새 URL로 치환합니다.
- **참고:**
  해당 함수는 코드 내에서 직접 호출되지는 않으나, 이미지 경로 변환이 필요한 경우 활용할 수 있습니다.

3.4. `report_generation(record_id: int)`

- **목적:**
  주어진 `record_id`(또는 CCTV ID)를 사용해 서버 API에서 관련 데이터를 GET 방식으로 가져옵니다.
- **동작:**
  - 기본 URL(`https://msteam5iseeu.ddns.net/api/person_count`)에 `record_id`를 결합하여 API 호출.
  - 요청에 성공하면 JSON 데이터를 파싱하여 반환하며, 실패 시 에러 메시지를 출력하고 `None` 반환.

3.5. `convert_html_to_pdf(pdf_file, member_id, cctv_id, report_title, persona, start_date, end_date)`

- **목적:**
  전체 보고서 생성을 위한 핵심 함수로, 데이터 수집, GPT 호출, HTML 저장, PDF 변환, 그리고 최종 서버 업로드를 처리합니다.
- **동작:**
  a. **데이터 수집:**
     - `report_generation(cctv_id)` 함수를 통해 CCTV ID에 해당하는 데이터를 가져옵니다.

b. **GPT 호출:**
   - `gpt_response` 함수를 호출하여, 지정된 페르소나, 날짜 범위 및 데이터를 기반으로 보고서를 생성하도록 요청합니다.

c. **HTML 저장:**
   - `save_html` 함수를 통해 GPT의 응답 중 HTML 콘텐츠를 추출하여 `response.html` 파일로 저장합니다.

d. **PDF 변환:**
   - `pdfkit` 라이브러리를 사용하여 저장된 HTML 파일을 PDF로 변환합니다.
   - wkhtmltopdf 실행 파일 경로와 옵션(인코딩, 페이지 사이즈, 여백 등)을 설정합니다.

e. **서버 업로드:**
   - 생성된 PDF 파일을 바이너리로 읽어, 멤버 ID, CCTV ID, 보고서 제목 등의 메타데이터와 함께 지정된 서버 URL(`https://msteam5iseeu.ddns.net/api/report`)로 POST 요청을 통해 전송합니다.

f. **예외 처리:**
   - 전 과정 중 발생할 수 있는 예외를 포착하여 오류 메시지를 출력하고 `None`을 반환하도록 구성합니다.

---

4. 실행 흐름 및 주의사항

- **테스트 호출:**
  코드의 마지막 부분에서 `convert_html_to_pdf` 함수가 테스트 목적으로 호출됩니다.
  - 매개변수로 PDF 파일 이름("aaa.pdf"), member_id(2), cctv_id(1), 보고서 제목("aaa"), 페르소나("돼지고기집"), 시작 날짜("2024-01-01"), 종료 날짜("2024-01-07")를 전달합니다.
- **경로 및 환경 설정:**
  - 실제 환경에서 데이터 파일 경로, wkhtmltopdf 실행 파일 경로, API 엔드포인트 등이 정확히 설정되어야 정상적으로 작동합니다.
  - 다수의 사용자 동시 사용 시, `response.html` 파일명 및 저장 경로의 충돌 가능성을 고려하여 별도의 파일 관리 방안이 필요합니다.

5. 한계 및 제한사항

5.1. 파일 및 경로 관리

- **고정 파일명 문제:**
  - 현재 HTML 파일(`response.html`)과 PDF 파일 이름이 고정되어 있어, 여러 사용자가 동시에 요청할 경우 파일 덮어쓰기나 충돌이 발생할 수 있습니다.
  - 파일 저장 경로 및 이름을 동적으로 생성하거나, 사용자별로 분리하는 관리 전략이 필요합니다.
- **로컬 이미지 경로:**
  - `convert_local_image_paths` 함수는 로컬 경로를 URL로 변환하지만, 서버 환경이나 배포 환경에서는 이미지 접근 권한 및 경로 설정에 따라 문제가 발생할 수 있습니다.

5.2. 외부 API 의존성

- **GPT 응답 품질:**
  - `gpt_response` 모듈의 응답 품질에 따라 보고서 내용의 정확성과 신뢰성이 좌우됩니다.
  - GPT 모델의 응답이 항상 일관적이지 않을 수 있으며, 특히 데이터 기반 보고서 생성 시 데이터와 맞지 않는 결과가 도출될 위험이 있습니다.
- **서버 API 호출:**
  - `report_generation` 함수와 PDF 업로드 시 사용하는 서버 API가 다운되거나 응답 지연이 발생할 경우 전체 파이프라인이 중단될 수 있습니다.
  - 타임아웃 및 예외 처리 로직이 존재하지만, 재시도 로직이나 장애 복구 전략은 추가적으로 고려해야 합니다.

5.3. 변환 도구 및 라이브러리 제한

- **pdfkit 및 wkhtmltopdf:**

- PDF 변환 과정은 wkhtmltopdf 실행 파일에 의존하므로, 해당 파일이 설치되어 있지 않거나 환경 변수 설정이 잘못될 경우 변환이 실패할 수 있습니다.
- PDF 변환 옵션(여백, 페이지 크기 등)은 고정되어 있으며, 다양한 출력 형식에 유연하게 대응하지 못할 수 있습니다.

5.4. 데이터 전처리 및 시각화

- **데이터 품질:**
  - 보고서 생성을 위한 데이터는 서버 API를 통해 가져오는데, 데이터의 정확성 및 최신성이 보장되지 않을 경우 보고서 내용의 신뢰도가 낮아질 수 있습니다.
- **시각화 제한:**
  - 코드 내에서는 데이터 전처리 및 시각화된 그래프를 보고서에 포함하는 부분이 암시적으로 포함되어 있으나, 실제 시각화 로직은 구현되어 있지 않습니다.
  - 향후 CV팀 등에서 제공하는 시각화 그래프를 동적으로 삽입하는 기능에 대한 별도 구현이 필요합니다.

5.5. 동시성 및 성능 문제

- **동시 사용자 처리:**
  - 단일 HTML 파일 및 PDF 파일 생성 방식은 다수의 요청을 동시에 처리할 때 성능 저하나 파일 접근 충돌의 원인이 될 수 있습니다.
  - 비동기 처리 또는 요청 큐 관리 등의 추가적인 성능 최적화가 필요할 수 있습니다.