1. 프로젝트 배경

본 프로젝트는 대피소 정보와 재난대피 요령을 제공하는 챗봇을 개발하는 것을 목표로 진행되었다. 공공데이터포털에서 제공하는 대피소 관련 CSV 데이터를 활용할 수 있고, 재난 행동요령 또한 텍스트 형태로 제공되어 이를 구조화하여 모델 학습에 사용할 수 있다는 점이 주요 선정 이유이다.

2. 데이터 수집 및 전처리 개요

재난 행동 요령 데이터는 사회재난 33개, 자연재난 25개로 구성되어 있으며, 그 중 대피소가 필요시되는 것으로 추려 사회재난 5개, 자연재난 8개로 진행하였다.

공공데이터포털에서 제공되는 대피소 CSV 파일은 원본 형태에서 행의 수가 18,630개, 열의 수가 26개로 구성된 비정형적 구조를 가지고 있었다. 행이 과도하게 많고, 도로명주소·도로명건물번호 등 중복되거나 활용도가 낮은 필드가 다수 포함되어 있었기 때문에, 분석 및 서비스 구현에 바로 활용하기 적합하지 않았다.

3. 불필요한 필드 제거 및 구조 정제

데이터 활용성을 높이기 위해 원본 CSV 파일의 행과 열들을 전수 검토하였으며, 이 중 중복 정보, 주소 세부 항목, 코드 반복 열 등을 중심으로 불필요한 필드를 제거하였다. 정제 작업 이후 행의 개수는 18,630개에서 17292개로 축소되었으며, 핵심적인 대피소 정보만 남도록 구조를 단순화하였다.열의 수는 추가적으로 제공하면 좋을만한 정보만 유지하여 총 13개로 진행하였다. 최종적으로 분석 및 시각화에 적합한 형태의 데이터셋으로 변환하였다.

4. 좌표 기반 데이터 활용

정제된 데이터에는 위도(latitude)와 경도(longitude) 값이 포함되어 있어, 이를 활용해 네이버 지도 API와 연동한 지도 시각화 작업을 수행하였다.이를 통해 사용자는 지도 기반으로 주변 대피소 위치를 직관적으로 확인할 수 있으며, 챗봇 또한 위치 정보에 기반한 응답을 제공할 수 있게 되었다.

5. 재난 행동요령 데이터 변환

재난 행동요령 데이터는 텍스트 형태로 제공되기 때문에, 챗봇이 효율적으로 활용할 수 있도록 필요한 항목을 추출하여 JSON 구조로 변환하였다.이 과정에서 각 재난 유형별로 행동 요령을 분류하고, 핵심 문장을 추출하여 구조화된 형식으로 정리하였다.

6. 문서(Document) 변환 및 청킹 처리 (documents.py)

LangChain 기반 RAG 모델 학습을 위해, 모든 CSV·JSON 데이터를 Document 객체로 변환하는 과정을 구현하였다.

1) CSV → Document 변환

- csv_to_documents()
- 대피소 데이터 각 행(row)을 하나의 Document로 변환
- page_content 예시:
- “민방위 대피 시설 OO은 ○○에 위치해 있으며, 최대 ○○명을 수용할 수 있습니다.”

metadata 구성 요소:

- 시설명, 시설구분, 주소, 운영상태, 관리번호

• 위도/경도 좌표
• 수용인원, 위치구분(지상/지하) 등
총 17,292개의 Document 생성
향후 지도 기반 응답 또는 시설 조건 검색에 유용하도록 메타데이터를 구조적으로 포함

2) JSON → Document 변환 (행동요령)
재난 행동요령 JSON은 깊은 트리 구조를 가지고 있어, 이를 파싱하기 위한 재귀 기반 구조를 설계하였다.
(1) parse_node(): JSON 재귀 파서
• JSON 구조의 모든 depth를 탐색하며 필요한 내용이 포함된 지점을 Document로 생성
• 주요 처리 항목:
• '세부사항', '주의사항', '내용', '이유', '신고처', '보호자 행동요령', '평소 준비사항', '행동요령'
breadcrumb 방식의 경로(path)를 포함하여 문맥을 보존
• 예: "지진 〉 대피요령 〉 실외에서의 행동 〉 주의사항"
추출된 항목을 page_content에 정리하며, 상황별 구분이 가능하도록 메타데이터에 situation, category, keyword 등을 넣어 구분
(2) json_to_documents(): 전체 JSON 처리
• 13개의 재난 행동요령 JSON을 일괄 처리하여 Document 생성
• 각 JSON의 상황(situation)을 기준으로 콘텐츠를 쪼개고 Document 리스트로 변환
• JSON 구조에 따라 수십~수백 개의 문서 생성
• 최종적으로 모든 JSON에서 수백 개 이상의 Document를 생성하여 RAG 적용 가능 상태로 만듦

7. 임베딩 생성 및 벡터 데이터베이스 구축

본 프로젝트에서는 재난 행동요령 텍스트와 대피소 관련 정보를 효율적으로 검색하고 활용할 수 있도록, LangChain과 OpenAI Embedding 모델을 활용하여 임베딩을 생성하고 벡터 데이터베이스(Vector DB)를 구축하였다.
임베딩 생성에는 OpenAI의 text-embedding-3-small 모델을 사용하였으며, 각 행동요령 문단 및 대피소 정보는 LangChain의 Document 형태로 가공하여 입력하였다. 생성된 임베딩은 Chroma DB를 기반으로 벡터 형태로 저장하였으며, 해당 데이터베이스는 추후 RAG 기반 질의응답(챗봇) 과정에서 주요 검색 인덱스로 활용된다.
OpenAI Embeddings 객체를 초기화하여 각 문서의 텍스트를 벡터 형태로 변환한 후에 Chroma Vector DB를 생성하고 문서와 임베딩 값을 저장한다.
저장된 벡터 DB는 로컬 경로(./chroma_db)에 영속적으로 보관되며, 재실행 시에도 동일한 인덱스를 사용할 수 있도록 구성한다.
이 과정을 통해 텍스트 기반 재난 행동요령과 대피소 데이터를 빠르게 검색할 수 있는 인덱스를 구축하였으며, 챗봇은 사용자 질의에 대해 보다 정확한 내용을 찾아 응답할 수 있다.

8. 전처리 및 벡터DB 구축 전체 흐름

본 프로젝트에서는 대피소 정보(CSV)와 다양한 자연·사회 재난 정보(JSON)를 하나의 검색 시스템에서 활용할 수 있도록 데이터를 통합하고, 이를 임베딩하여 벡터 데이터베이스로 구축하였다. 이 과정은 모두 모듈화된 구조로 설계되어 유지보수성과 확장성을 높였다.

8.1 Data Load
서로 다른 형식의 원본 데이터를 불러오는 단계가 진행된다.
대피소 데이터(CSV): 행정구역, 대피소명, 주소, 위도·경도 등 위치 기반 정보가 포함되어 있다.
재난 데이터(JSON): 태풍, 홍수, 지진, 화산폭발, 산불, 방사능 등 총 13종의 자연·사회 재난 정보를 포함한다.
각 파일은 전용 로딩 함수를 통해 표준화된 형태의 파이썬 객체로 변환되었다.

8.2 Document 변환
로딩된 데이터는 검색 가능한 구조로 만들기 위해 모두 Document 형태로 변환하였다.
JSON 재난 데이터는 재난의 정의, 설명, 대응 방법 등의 텍스트 중심 정보를 기반으로 Document로 구성되었으며,
총 150개의 Document가 생성되었다.
CSV 대피소 데이터는 대피소의 정보 및 위치 데이터를 바탕으로 Document로 변환되었으며,
총 17,272개의 Document가 생성되었다.
이 과정에서 모든 데이터는 일관된 문서 구조를 갖도록 표준화하여 이후 임베딩 과정에 적합하게 준비하였다.

8.3 문서 통합 및 임베딩 생성
Document 형태로 변환된 재난 정보(150개)와 대피소 정보(17,272개)를 모두 통합하여
총 17,422개의 전체 문서 집합을 구성하였다.
이 통합 문서를 기반으로 텍스트 임베딩 모델을 사용하여 벡터 형태로 변환하였다.
이를 통해 각 문서는 의미 기반 검색이 가능한 수치 표현 벡터로 변환된다.

8.4 벡터 데이터베이스(Vector DB) 구축
임베딩이 완료된 17,422개의 문서는 벡터 데이터베이스로 저장되었다.
벡터 DB는벡터 인덱싱, 의미 기반 유사도 검색, 빠른 검색 속도 확보 기능을 제공한다.
그 결과, 실행 로그에 따라
“vectordb 생성 완료: 17,422개 문서 저장”
이 출력되며 벡터 DB 구축이 성공적으로 완료되었음을 확인하였다.

8.5 전체 과정 요약
본 프로젝트의 데이터 전처리 및 벡터DB 구축 전체 흐름을 요약하면 다음과 같다.
1. CSV 및 JSON 데이터 수집 및 로드
2. 각 데이터를 Document 형태로 변환하여 표준화
3. 모든 문서를 통합하여 일관된 데이터셋 구성
4. 텍스트 임베딩 생성
6. 벡터 데이터베이스 구축 및 인덱싱 완료

이를 통해 검색 기반 재난 정보 서비스의 기반 구조가 완성되었다.