수집된 데이터 및 데이터 전처리 문서

1. 데이터 수집

1-1 데이터 소스

목적	설명	출처
이력서 데이터	22년도 인프런에서 런칭한 커리어 hub형 플랫폼	랠릿
회사 데이터	theVC: 16년도 런칭한 스타트업 및 투자기관 자료 전문 DB형 플랫폼	더브이씨

1-2 데이터 수집 방법

- selenium을 통한 크롤링
    - 랠릿 허브 접속
    - 각 허브에 있는 이력서 원문 수집 및 PDF파일로 저장

1-3 수집된 데이터 요약

데이터 이름	크기 (MB/행 수)	데이터 형태	수집 날짜
이력서 데이터	1.29GB / 3240	TXT	24.12.30
회사 데이터	254KB / 2356	TXT	24.01.02

2. 데이터 전처리

2-1 작업목표

- 원문 형태로 저장된 TXT 이력서 데이터를 JSON 형태로 변환 및 정규화.
- 변환된 데이터를 AWS S3에 저장하고, 이후 DB에 적재하여 활용 가능하도록 준비.

2-2 사용 도구 및 API

- OpenAI API: 프롬프트 엔지니어링을 통해 이력서 데이터를 JSON 형식으로 변환
- AWS S3: 정규화된 JSON 데이터를 저장
- DB: 최종 데이터를 적재 및 관리

2-3 워크플로우

2-3-1 OpenAI API를 활용한 데이터 전처리

- 프롬프트 설계
  - 이력서 데이터를 JSON 형식으로 구조화하기 위해 적절한 지시사항 설계.

사용 프롬프트

```python
from openai import OpenAI
from tqdm import tqdm
from dotenv import load_dotenv
import os
# .env 파일에서 환경 변수 로드
load_dotenv()

# OpenAI API 키를 환경 변수에서 가져옴
api_key = os.getenv("OPENAI_API_KEY")

class ResumePreprocessor:
    def __init__(self, batch_size=1):
        self.client = OpenAI(api_key=api_key)
        self.batch_size = batch_size

    def process_resume(self, resume_data):
        # 단일 이력서 처리
        completion = self.client.chat.completions.create(
            model='gpt-4o-mini',
            messages=[
            {
                'role': 'system',
                'content': '''너는 관련된 내용을 아래 json 형식에 맞춰 바꿔주는 봇이야
                {
                    "Profile": {
                        "name": "프로필 이름 (예: 김철수)",
                        "job_category": "직업 카테고리 (예: 소프트웨어 엔지니어, 데이터 사이언티스트 )",
                        "career_year": "경력 연수 (정수 값, 예: 5, 직업 카테고리로 쌓인 경력만을 활용해줘)"
                    },
                    "TechStack": [
                        {
                        "tech_stack_name": "기술 스택 이름 (예: Python, Django, React)"
                        }
                    ],
                    "Career": [
                        {
                        "company_name": "회사명 (예: 테크코퍼레이션)",
                        "position": "직위 (예: Senior Software Engineer, Software Developer, 하나의 직위만 표현하
                        "start_date": "시작일 (YYYY-MM 형식, 예: 2018-01)",
                        "end_date": "종료일 또는 null (YYYY-MM 형식, 예: 2022-12)",
                        "is_currently_employed": "현재 재직 여부 (Boolean 값, 예: True 또는 False)",
                        "responsibilities": "담당 업무 (예: 웹 애플리케이션 개발 및 유지보수)",
                        "description": "추가 설명 (예: 글로벌 팀 프로젝트를 진행)"
```

```
                        }
                    ],
                    "AcademicRecord": [
                        {
                        "school_name": "학교명 (예: 서울대학교)",
                        "major": "전공 (예: 컴퓨터공학)",
                        "status": "졸업 상태 (예: 졸업, 재학, 중퇴)",
                        "enrollment_date": "입학일 (YYYY-MM 형식, 예: 2014-09)",
                        "graduation_date": "졸업일 또는 null (YYYY-MM 형식, 예: 2018-06)"
                        }
                    ],
                    "Certificate": [
                        {
                        "name": "자격증 이름 (예: 정보처리기사, ADsP, 빅데이터 분석기사, 외국어 관련 자격증은 제외해줘)
                        }
                    ],
                    "Language": [
                        {
                        "language_name": "언어 이름 (예: 영어, 스페인어)",
                        "description": "이력서의 내용으로 해당 언어 능력을 판단해서 꼭 판단 근거와 함께 길게 작성해줘"
                        }
                    ]
                }
                key 에 대응되는 값이 없을경우 key는 남겨두고 value에는 null값을 넣어줘
                '''
            },
            {
                'role': 'user',
                'content': resume_data
            }
        ],
        temperature=0.4
        )

        # API 응답에서 변환된 결과를 추출
        result = completion.choices[0].message.content
        return result

```

- 사용 모델: gpt-4o
  - temperture: 0.4

2-4 AWS S3 데이터 DB 적재

- **데이터 가져오기**:
  - AWS SDK 또는 CLI를 사용하여 S3에서 JSON 데이터 다운로드.
- **DB 스키마 설계**:
  - 테이블 이름: Profile
  - 주요 컬럼: 이름, 연락처, 이메일, 경력, 학력, 기술 스택.