📊 데이터 수집 및 전처리 문서

목차

1. 개요
2. 데이터 수집
3. 데이터 전처리
4. 최종 데이터 구조
5. 데이터 통계

개요

본 프로젝트는 **온통청년 API**를 통해 청년 정책 데이터를 수집하고, RAG 시스템에 적합한 형태로 전처리하여 벡터 데이터베이스에 저장합니다.

데이터 파이프라인

```
온통청년 API → 원본 JSON → 전처리 → 벡터화 → ChromaDB
```

---

1. 데이터 수집

1.1 데이터 소스

- **출처**: 온통청년 정책 포털 (https://www.youthcenter.go.kr)
- **API 엔드포인트**: `https://www.youthcenter.go.kr/go/ythip/getPlcy`
- **인증 방식**: API Key (환경 변수: `YOUTH_POLICY_API`)

1.2 수집 스크립트

- **파일**: `notebooks/fetch_api_data.py`
- **주요 기능**:
  - API 요청 및 응답 처리
  - JSON 데이터 검증
  - 원본 데이터 저장

1.3 API 요청 파라미터

```
{
    'apiKeyNm': YOUTH_POLICY_API,
    'pageSize': 3550  # 한번에 가져올 정책 개수
}
```

1.4 원본 데이터 파일

- **경로**: `data/raw/youth_policies_api.json`
- **파일 크기**: 11.71 MB
- **실제 수집**: 3,550개 정책 (pageSize 제한)

1.5 원본 데이터 구조


```
{
  "resultCode": 200,
  "resultMessage": "성공적으로 데이터를 가지고 왔습니다.",
  "result": {
    "pagging": {
      "totCount": 4301,
      "pageNum": 1,
      "pageSize": 3550
    },
    "youthPolicyList": [
      {
        "plcyNo": "정책번호",
        "plcyNm": "정책명",
        "plcyExplnCn": "정책설명",
        "lclsfNm": "대분류",
        "mclsfNm": "중분류",
        "plcySprtCn": "지원내용",
        "sprtTrgtMinAge": "최소연령",
        "sprtTrgtMaxAge": "최대연령",
        "sprvsnInstCdNm": "주관기관명",
        "rgtrInstCdNm": "등록기관명",
        "aplyYmd": "신청기간",
        "bizPrdBgngYmd": "사업시작일",
        "bizPrdEndYmd": "사업종료일",
        "refUrlAddr1": "참고URL",
        // ... 약 60개 필드
      }
    ]
  }
}
```


---

2. 데이터 전처리

2.1 전처리 목표

1. **필드 정제**: 필요한 필드만 선택 (60개 → 28개)
2. **한글 필드명**: 영문 코드를 한글로 변환 (가독성 향상)
3. **데이터 검증**: Null, 빈 값 처리
4. **지역 정보 추출**: 기관명에서 지역 추출
5. **날짜 포맷 통일**: YYYYMMDD 형식 유지

2.2 전처리 프로세스

Step 1: 필드 매핑

원본 영문 필드를 한글 필드로 변환:

원본 필드	전처리 필드	설명
plcyNm	정책명	정책 이름
plcyKywdNm	정책키워드	검색 키워드
plcyExplnCn	정책설명	상세 설명
lclsfNm	대분류	정책 카테고리 (대)
mclsfNm	중분류	정책 카테고리 (중)
plcySprtCn	지원내용	지원 혜택
earnMinAmt	최소지원금액	지원금 최소
earnMaxAmt	최대지원금액	지원금 최대
sprtTrgtMinAge	지원최소연령	대상 연령 (최소)
sprtTrgtMaxAge	지원최대연령	대상 연령 (최대)
sprvsnInstCdNm	주관기관명	정책 주관 기관
rgtrInstCdNm	등록기관명	정책 등록 기관
aplyYmd	신청기간	신청 가능 기간
bizPrdBgngYmd	사업시작일	사업 시작일
bizPrdEndYmd	사업종료일	사업 종료일
refUrlAddr1	참고URL1	상세 정보 링크

Step 2: 코드 값 변환

API의 코드 값을 의미있는 텍스트로 변환:

제공기관그룹 (pvsnInstGroupCd)

```
{
    "0054001": "중앙부처",
    "0054002": "지자체",
    "0054003": "공공기관",
    "0054010": "제한없음"
}
```

정책제공방법 (plcyPvsnMthdCd)

```
{
    "0042001": "현금",
    "0042002": "현물",
    "0042003": "서비스",
    "0042006": "보조금",
    "0042010": "제한없음"
}
```

혼인상태 (`mrgSttsCd`)

```
{
    "0055001": "미혼",
    "0055002": "기혼",
    "0055003": "제한없음"
}
```

Step 3: 지역 코드 → 한글 변환

법정동코드를 한글 지역명으로 변환:

법정동코드 매핑 파일: `data/processed/법정동코드 수정.txt`

- **총 280개** 시/도, 시/군/구 매핑 테이블
- 형식: 법정동코드 (TAB) 법정동명

```
# 변환 예시
"11110" → "서울특별시 종로구"
"26110" → "부산광역시 중구"
"48170" → "경상남도 남해군"

# 다중 지역 코드 처리
"11110,11140,11170" → "서울특별시 종로구, 서울특별시 중구, 서울특별시 용산구"
```

변환 프로세스:

1. 원본 JSON의 `zipCd` 필드 (쉼표로 구분된 법정동코드)
2. 각 코드를 법정동코드 매핑 테이블과 매칭
3. 한글 지역명으로 변환하여 `지역` 필드 생성
4. 여러 코드가 있을 경우 쉼표로 연결

Step 4: 데이터 검증 및 정제

```
# Null 값 처리
if value is None or value == "":
    value = "정보없음"
```

```
# 숫자 필드 검증
if not value.isdigit():
    value = "0"

# 날짜 형식 검증
if not re.match(r'^\d{8}$', date_value):
    date_value = None
```

2.3 전처리 결과 파일

- **경로**: `data/processed/youth_policies_filtered_kr_revised.json`
- **파일 크기**: 11.29 MB
- **총 레코드 수**: 3,550개 정책

---

3. 최종 데이터 구조

3.1 JSON 스키마


```
[
  {
    "정책명": "string",
    "정책키워드": "string",
    "정책설명": "string",
    "대분류": "string",
    "중분류": "string",
    "지원내용": "string",
    "최소지원금액": "string",
    "최대지원금액": "string",
    "기타지원조건": "string (optional)",
    "지원최소연령": "string",
    "지원최대연령": "string",
    "주관기관명": "string",
    "등록기관명": "string",
    "신청기간": "string (optional)",
    "사업시작일": "string",
    "사업종료일": "string",
    "참고URL1": "string",
    "재공기관그룹": "string",
    "정책제공방법": "string",
    "정책승인상태": "string",
    "신청기간구분": "string",
    "사업기간구분": "string",
    "혼인상태": "string",
    "소득조건": "string",
    "전공요건": "string",
    "취업상태": "string",
    "학력요건": "string",
    "특화분야": "string",
    "지역": "string"
```

```
  }
]
```

3.1.1 추가 메타데이터

- 지역범위 : 특정 지역필드에서 지역이 전부 적힌 경우 "전국", 일부만 적힌 경우 "지역"로 구분

```
{
    "지역": "경기도",
    "지역범위": "지역"
}
# 시도 목록
SIDO_LIST = [
    "서울특별시","부산광역시","대구광역시","인천광역시","광주광역시",
    "대전광역시","울산광역시","세종특별자치시","경기도","강원특별자치도",
    "충청북도","충청남도","전북특별자치도","전라남도","경상북도",
    "경상남도","제주특별자치도"
]
# "지역"필드에 시도명이 모두 포함된 경우 "전국"으로 설정
{
  "지역" :"서울특별시 종로구, 서울특별시 중구, 서울특별시 용산구, 서울특별시 성동
구,...",
  "지역범위": "전국"
}
```

3.2 필드 설명

기본 정보 (6개)

- 정책명: 정책의 공식 명칭
- 정책키워드: 검색용 키워드 (쉼표 구분)
- 정책설명: 정책의 상세 설명 (개요, 내용)
- 대분류: 정책 유형 대분류 (일자리, 주거, 교육, 복지문화, 참여권리)
- 중분류: 정책 유형 중분류
- 지원내용: 실제 지원 혜택 및 방법

지원 조건 (8개)

- 최소지원금액: 지원금 최소 금액 (원)
- 최대지원금액: 지원금 최대 금액 (원)
- 기타지원조건: 추가 지원 조건 (소득, 자산 등)
- 지원최소연령: 지원 대상 최소 연령
- 지원최대연령: 지원 대상 최대 연령
- 혼인상태: 미혼/기혼/제한없음
- 소득조건: 소득 요건 (무관/제한있음)
- 학력요건: 학력 제한 사항

기관 및 일정 (7개)

- 주관기관명: 정책 주관 기관
- 등록기관명: 정책 등록 기관
- 신청기간: 신청 가능 기간 (YYYYMMDD ~ YYYYMMDD)
- 사업시작일: 사업 시작일 (YYYYMMDD)
- 사업종료일: 사업 종료일 (YYYYMMDD)
- 신청기간구분: 상시/특정기간/미정
- 사업기간구분: 상시/특정기간/미정

추가 정보 (7개)

- 참고URL1: 상세 정보 링크
- 재공기관그룹: 중앙부처/지자체/공공기관
- 정책제공방법: 현금/현물/서비스/보조금
- 정책승인상태: 승인/반려/검토중
- 전공요건: 전공 제한 사항
- 취업상태: 재직/미취업/제한없음
- 특화분야: 특정 분야 제한
- 지역: 정책 시행 지역 (시/도 + 시/군/구)

---

4. 데이터 통계

4.1 기본 통계

항목	값
총 정책 수	3,550개
원본 파일 크기	11.71 MB
전처리 파일 크기	11.29 MB
필드 수	28개 (원본 60개 → 28개)

4.2 정책 카테고리 분포 (대분류)

```
일자리:      약 1,200개 (33.8%)
주거:        약 800개  (22.5%)
교육:        약 700개  (19.7%)
복지문화:    약 600개  (16.9%)
참여권리:    약 250개  (7.0%)
```

4.3 제공기관 분포

```
지자체:      약 2,500개 (70.4%)
중앙부처:    약 800개  (22.5%)
```

```
공공기관:    약 250개  (7.0%)
```

4.4 연령대 분포

```
19-24세:     약 1,200개 (33.8%)
25-29세:     약 1,400개 (39.4%)
30-34세:     약 800개  (22.5%)
35-39세:     약 150개  (4.2%)
```

4.5 지역 분포 (상위 10개)

```
1. 서울특별시:     약 500개 (14.1%)
2. 경기도:         약 450개 (12.7%)
3. 부산광역시:     약 300개 (8.5%)
4. 대구광역시:     약 250개 (7.0%)
5. 인천광역시:     약 200개 (5.6%)
6. 광주광역시:     약 180개 (5.1%)
7. 대전광역시:     약 170개 (4.8%)
8. 울산광역시:     약 150개 (4.2%)
9. 경상남도:       약 140개 (3.9%)
10. 전라남도:      약 130개 (3.7%)
```

4.6 정책 유형 분포 (중분류 상위 5개)

```
1. 취업지원:          약 600개 (16.9%)
2. 창업지원:          약 500개 (14.1%)
3. 주거금융지원:      약 450개 (12.7%)
4. 생활복지지원:      약 400개 (11.3%)
5. 교육·훈련:         약 350개 (9.9%)
```

---

5. 벡터화 및 저장

5.1 벡터 데이터베이스

- **DB 유형**: ChromaDB
- **경로**: `data/vectordb/`
- **총 크기**: 약 87 MB

5.2 벡터화 설정

- **임베딩 모델**: OpenAI `text-embedding-3-small`
- **벡터 차원**: 1,536차원
- **총 벡터 수**: 3,550개

- **벡터 파일**: `data_level0.bin` (17.98 MB)

5.3 메타데이터 저장

- **DB 파일**: `chroma.sqlite3` (69.38 MB)
- **저장 내용**:
  - 원본 텍스트 (정책 정보)
  - 메타데이터 (28개 필드)
  - 벡터 ID 매핑

5.4 인덱스 구조

- **알고리즘**: HNSW (Hierarchical Navigable Small World)
- **인덱스 파일**: `link_lists.bin`, `header.bin`, `length.bin`
- **검색 성능**: O(log n) 시간 복잡도

---

6. 데이터 품질 관리

6.1 데이터 검증 항목

- ☑ 필수 필드 존재 여부 (정책명, 지원내용, 기관명 등)
- ☑ 날짜 형식 검증 (YYYYMMDD)
- ☑ 숫자 필드 검증 (연령, 금액)
- ☑ URL 유효성 검증
- ☑ 중복 정책 제거
- ☑ 빈 값 처리 (기본값 설정)

6.2 데이터 정제 규칙

```
# 1. Null 값 처리
null_value → "정보없음"

# 2. 금액 필드
비어있음 → "0"
음수 → "0"

# 3. 연령 필드
비어있음 → "0"
범위 초과 → "0" ~ "99"

# 4. 날짜 필드
비어있음 → None
형식 오류 → None

# 5. URL 필드
비어있음 → ""
형식 오류 → 원본 유지
```

---

7. 참고 자료

7.1 관련 파일

- `notebooks/fetch_api_data.py`: 데이터 수집 스크립트
- `notebooks/build_vectordb.py`: 벡터 DB 구축 스크립트
- `data/raw/youth_policies_api.json`: 원본 데이터
- `data/processed/youth_policies_filtered_kr_revised.json`: 전처리 데이터

---

8. 트러블 슈팅

문제 요약

1. 불필요한 필드 삭제

초기 전처리 단계에서 불필요하다고 판단한 여러 원본 필드를 삭제. 최초 전처리에서 삭제한 컬럼 목록은 다음과 같습니다:

- `['plcyNo', 'bscPlanCycl', 'bscPlanPlcyWayNo', 'bscPlanFcsAsmtNo', 'bscPlanAsmtNo', 'plcyPvsnMthdCd', 'plcyAprvSttsCd', 'sprtSclLmtYn', 'bizPrdEtcCn', 'etcMttrCn', 'sprtSclCnt', 'sprtArvlSeqYn', 'sprtTrgtAgeLmtYn', 'mrgSttsCd', 'inqCnt', 'zipCd', 'plcyMajorCd', 'jobCd', 'schoolCd', 'sbizCd']`

초기 전처리 후 유지된 주요 컬럼(검사 시점 기준)은 다음과 같습니다:

- `['pvsnInstGroupCd', 'plcyNm', 'plcyKywdNm', 'plcyExplnCn', 'lclsfNm', 'mclsfNm', 'plcySprtCn', 'sprvsnInstCd', 'sprvsnInstCdNm', 'sprvsnInstPicNm', 'operInstCd', 'operInstCdNm', 'operInstPicNm', 'aplyPrdSeCd', 'bizPrdSeCd', 'bizPrdBgngYmd', 'bizPrdEndYmd', 'plcyAplyMthdCn', 'srngMthdCn', 'aplyUrlAddr', 'sbmsnDcmntCn', 'refUrlAddr1', 'refUrlAddr2', 'sprtTrgtMinAge', 'sprtTrgtMaxAge', 'earnCndSeCd', 'earnMinAmt', 'earnMaxAmt', 'earnEtcCn', 'addAplyQlfcCndCn', 'ptcpPrpTrgtCn', 'rgtrInstCd', 'rgtrInstCdNm', 'rgtrUpInstCd', 'rgtrUpInstCdNm', 'rgtrHghrkInstCd', 'rgtrHghrkInstCdNm', 'aplyYmd', 'frstRegDt', 'lastMdfcnDt']`

- 추가 점검

추가적인 점검 과정에서 중복 처리 및 확인 누락으로 인해 일부 필드가 불필요하게 삭제된 것이 확인되어 다음 필드들을 삭제 처리했다고 기록합니다:

- 모든 ~코드 (법정동/지역 코드 포함; 이 과정에서 지역 코드가 함께 제거되어 데이터에서 지역 컬럼이 누락되었습니다)
- 최초등록일 (frstRegDt)
- 최종수정일 (lastMdfcnDt)
- 담당부서명
- 담당자명

- 이후 수정 사항

이후 데이터 검토 과정에서 지역 필드가 누락된 것을 발견, 해당 필드를 복원하기 위해 법정동코드 매핑을 재적용하였으며, 최초등록일 및 최종수정일 필드는 데이터 분석에 불필요하다고 판단하여 삭제를 유지하였습니다.

2. 기관명으로 추정한 지역 정보의 문제

초기 전처리에서는 주관기관명, 등록기관명, 상위기관명 필드의 값을 기반으로 지역을 추정하는 로직을 사용했습니다. 그러나 많은 주관기관은 특정 지역에 소재하더라도 사업을 전국 단위로 운영하거나 타지역 참여가 가능한 경우가 많아, 기관명을 그대로 지역으로 매핑하면 다음과 같은 오류가 발생했습니다:

- 특정 기관명이 포함된 정책을 해당 기관의 소재지 지역으로 잘못 분류하여 지역 필터링 시 해당 정책이 제외됨
- 실제로는 전국 대상임에도 불구하고 지역별 검색 결과에서 누락되는 사례 발생

- 이후 수정 사항

- 주관기관명 등 기관명만으로 지역을 단정하지 말고, 우선 지역범위(전국/지역) 필드를 우선 사용.
- 법정동 코드(`zipCd`) 또는 명시적 지역 필드를 우선 매핑하고, 기관명 기반 추정은 보조 수단으로만 활용.
