

테스트 및 결과 보고서

팀명: 청바지(1Team)

팀원: 강민지, 나호성, 이지은, 조준상, 홍혜원

목차

1. 전체 파이프라인 개요.....	2
1.1 전체 파이프라인 흐름 (End-to-End Flow).....	2
1.2 검색 및 필터링 구조.....	2
2. Class 단위 상세 설명 (Class Specifications).....	3
2.1 QueryRouter	4
2.2 RegionFilter	4
2.3 MultiQueryGenerator.....	4
2.4 EnsembleRetriever.....	5
2.5 ReciprocalRankFusion (RRF)	5
2.6 AdvancedRAGPipeline	6
3. 테스트 설계	7
3.1 테스트 환경	7
3.2 테스트 시나리오 및 케이스	7
4. 문제점 및 해결과정	12
5. 결론 및 향후 과제	14
5.1 최종 평가.....	14
5.2 향후 과제	14

1. 전체 파이프라인 개요

본 문서는 청년 정책 정보 제공을 위한 **고급 RAG(Retrieval-Augmented Generation) 시스템**의 아키텍처, 구성 요소, 그리고 테스트 결과를 기술한다. 본 시스템은 사용자의 자연어 질문을 이해하고, 방대한 정책 문서 중 가장 적합한 정보를 검색하여, 신뢰할 수 있는 답변을 생성하는 것을 목표로 한다.

1.1 전체 파이프라인 흐름 (End-to-End Flow)

시스템은 사용자의 질문이 입력되는 순간부터 최종 답변이 생성되기까지 다음 8단계의 프로세스를 거친다.

1. **입력 (Input):** 사용자가 자연어로 정책 관련 질문을 입력한다.
2. **라우터 (Router):** 질문의 유효성을 검증하고, 의도(Intent)를 파악하여 검색이 필요한 질문인지 분류한다.
3. **지역 추출:** 질문에 포함된 지역 정보(예: "대구", "전국")를 LLM을 통해 추출하고 메타데이터 필터를 생성한다.
4. **쿼리 확장:** 단일 질문을 다양한 관점의 검색 쿼리(Multi-Query)로 확장하여 검색 커버리지를 높인다.
5. **하이브리드 검색(Retrieval):**
 - A. Dense Retrieval: 의미 기반 검색 (Vector Store / ChromaDB)
 - B. Sparse Retrieval: 키워드 기반 검색 (BM25)
6. **결과 통합 (RRF Fusion):** Reciprocal Rank Fusion 알고리즘을 사용하여 서로 다른 검색 전략의 결과를 재순위화(Re-ranking)하고 통합한다.
7. **후처리 (Post-Filtering):** 통합된 결과 중 사용자의 지역 조건과 일치하지 않는 문서를 최종 필터링한다.
8. **생성:** 검색된 문서(Context)와 대화 기록(Memory)을 바탕으로 LLM이 최종 답변과 요약본을 생성한다.

1.2 검색 및 필터링 구조

- Chroma DB

정책 문서의 임베딩 벡터와 메타데이터(지역, 분류 등)를 저장하는 벡터스토어다.

- **EnsembleRetriever**

Dense Retriever(Vector)와 Sparse Retriever(BM25)를 병렬로 실행하여 의미적 유사성과 키워드 일치성을 동시에 확보한다.

- **Metadata Filtering**

- **Pre-Filtering**

사용자 입력에 '전국', '전체'와 같은 단어가 있을 시, 'filter={"지역범위": "전국"}'을 적용하여 검색 범위를 좁힌다.

사용자 입력에 '서울', '대구'와 같은 특정 지역명이 있을 시, 필터를 적용하지 않고 None을 반환한다.

```
AdvancedRAGPipeline.query()
↓
RegionFilter.build_filter() → metadata_filter 생성
↓
AdvancedRAGPipeline.query()에서 metadata_filter 보유
↓
EnsembleRetriever.retrieve(queries, metadata_filter) 호출
↓
EnsembleRetriever.dense_search(query, metadata_filter) 호출
↓
ChromaDB에 metadata_filter 적용하여 검색
```

- **Post-Filtering**

사용자가 입력에 특정 지역명이 있을 시, 사용자의 요구(특정 지역 + 전국)에 따라 내부 로직에서 해당 지역을 필터링한다.

```
filtered_docs = []
for doc in documents:
    # 전국 정책은 항상 포함
    if doc.metadata.get('지역범위') == '전국':
        filtered_docs.append(doc)
    # 지역 필드에 해당 지역명이 포함되어 있으면 포함
    elif region_name in doc.metadata.get('지역', ''):
        filtered_docs.append(doc)

return filtered_docs if filtered_docs else documents
```

2. Class 단위 상세 설명 (Class Specifications)

2.1 QueryRouter

2.1.1 사용 이유

모든 입력에 대해 무조건 검색을 수행하면 리소스가 낭비되고 무의미한 질문 (인사, 욕설 등)에 대해 엉뚱한 정책을 검색하는 환각현상이 발생할 수 있다.

2.2.2 목적

입력된 텍스트가 정책 질문으로서 유효한지 판단하고 검색에 적합한 형태로 정제한다.

2.1.3 작동 순서

1. 사용자 쿼리를 입력 받는다.
2. LLM에게 프롬프트를 통해 유효성, 카테고리, 정제된 쿼리를 요청한다.
3. JSON 형식으로 응답한다.
4. 유효하지 않은 질문은 즉시 거절 메시지를 반환하고, 유효한 경우 쿼리를 다음 단계로 넘긴다.

2.2 RegionFilter

2.2.1 사용 이유

청년 정책은 '거주지'가 가장 중요한 자격 요건이다. 단순 키워드 검색만으로는 "대구 청년 정책"을 물었을 때 "서울 청년 정책"이 검색되는 문제를 막기 어렵다.

2.2.2 목적

자연어 질문에서 지역 정보를 명확히 추출하고 이를 데이터베이스 필터링 조건으로 변환한다.

2.2.3 작동 순서

1. ``detect_region(query)``: LLM을 이용해 질문에서 지역명(예: 대구)과 전국 여부(is_national)를 추출한다.
2. ``build_filter(region_info)``: 추출된 정보를 바탕으로 ChromaDB용 메타데이터 필터(예: ``{"지역범위": "전국"}``)를 생성한다.
3. ``filter_documents(docs, region_info)``: 검색이 완료된 문서 리스트를 순회하며 사용자가 요청한 지역과 문서의 메타데이터가 일치하지 않는 문서를 제거한다.

2.3 MultiQueryGenerator

2.3.1 사용 이유

사용자의 질문이 모호하거나 특정 단어에 편향될 수 있다. 단일 쿼리 검색은 해당 단어가 없는 관련 문서를 놓칠 수 있다.

2.3.2 목적

하나의 질문을 다양한 관점과 표현을 가진 여러 개의 쿼리로 확장하여 검색 범위를 넓힌다.

2.3.3 작동 순서

1. 원본 쿼리를 입력 받는다.
2. LLM에게 3가지 관점(지역 강조, 정책 키워드 중심, 유사어 포함)의 쿼리 생성을 요청한다.
3. 생성된 쿼리들을 리스트로 변환하여 원본 쿼리와 합친다.
4. 확장된 쿼리 리스트를 검색기에 전달한다.

2.4 EnsembleRetriever

2.4.1 사용 이유

Vector 검색(Dense)은 의미 파악에 강하지만 고유명사나 정확한 키워드 매칭에 약하다. 반면 BM25(Sparse)는 키워드 매칭에 강하다. 둘을 결합해 상호 보완이 필요하다.

2.4.2 목적

서로 다른 알고리즘을 병렬로 실행하여 최적의 문서 후보를 수집한다.

2.4.3 작동 순서

1. `_build_bm25`, `_build_vector`를 통해 각 Retriever를 초기화한다.
2. `retrieve(queries)`가 호출되면 입력된 모든 쿼리에 대해 반복문을 수행한다.
3. 각 쿼리마다 Vector Search와 BM25 Search를 독립적으로 수행한다.
4. 수집된 모든 결과(중복 포함)를 리스트 형태로 반환한다.

2.5 ReciprocalRankFusion (RRF)

2.5.1 사용 이유

Dense와 Sparse 검색 결과의 점수 스케일이 다르기 때문에(유사도 vs BM25 점수), 단순 합산으로는 공정한 순위를 매기기가 어렵다.

2.5.2 목적

순위(Rank) 기반으로 점수를 재산정하여, 여러 검색기에서 공통적으로 상위 점수의 문서를 최상단으로 올린다.

2.5.3 작동 순서

1. EnsembleRetriever로부터 받은 결과 딕셔너리를 순회한다.
2. 각 문서의 순위를 확인하고 $score = 1 / (k + rank)$ 공식을 적용하여 점수를 부여한다 (k=60).
3. 동일 문서가 여러 검색 결과에 등장하면 점수를 누적한다.
4. 최종 점수순으로 정렬하여 상위 K개(Top-K) 문서만 남긴다.

2.6 AdvancedRAGPipeline

2.6.1 사용 이유

위에서 정의한 개별 요소들을 유기적으로 연결하고 데이터 흐름을 제어할 중앙 컨트롤러가 필요하다.

2.6.2 목적

사용자 입력부터 최종 응답까지의 전체 프로세스를 운영한다.

2.6.2 작동 순서

1. `query()` 메서드가 호출되면 *Router* → *RegionFilter* → *MultiQuery* → *Ensemble* → *RRF* → *Post-Filter* 순으로 데이터를 통과시킨다.
2. 최종 선별된 문서들을 텍스트로 변환하여 프롬프트에 삽입한다.
3. Memory에서 대화 맥락을 가져온다.
4. LLM을 호출하여 답변과 요약을 생성하고 반환한다.

3. 테스트 설계

3.1 테스트 환경

- **OS:** Windows 11
- **Python:** 3.13.9
- **LLM:** GPT-4o-mini (Temperature=0.0)
- **Embedding:** text-embedding-3-small
- **Vector DB:** ChromaDB
- **Data:** 청년 정책 데이터(온통청년API) 3,550건

3.2 테스트 시나리오 및 케이스

3.2.1 지역 필터링 정확도 테스트

질문에 포함된 지역 정보를 정확히 인식하고, 타 지역 정책을 배제하는지 확인한다.

➤ Case 1 지역 명시

"서울 취업 지원" -> 서울특별시 정책 + 전국대상 정책만 포함 (타지역 제외)

■전체 답변:
안녕! 나는 청년들의 든든한 정책 선배, 청년이름 선배봇 🌟이야.
주거, 월세, 일자리, 복지 정책 등 궁금한 점이 있으면 언제든지 나에게 물어봐! 🐾

사용자 질문 : 서울 취업 지원

답변 :

1. 서울 청년 영상취재단 모집

- ◆사업 개요
 - 사업 기간 : 2024년 7월 ~ 2024년 12월
 - 목적 : 서울시에서 추진하는 다양한 정책·행사 등을 시민 눈높이에 맞게 쉽고 재미있는 숏폼 영상으로 제작할 수 있는
- ◆신청 자격(핵심 요건)
 - 연령 : 청년
 - 주거 : 서울특별시
 - 소득 : 제한 없음
 - 기타 조건 : 청년이 운영하는 유튜브·인스타그램·기타 채널로서 시에서 추진하는 정책·사업·행사 등을 숏폼 영상으로
- ◆지원 금액·기간
 - 월 지원 금액 : 숏폼 1편당 55만원
 - 지원 기간 : 6개월
- ◆신청 방법(절차)
 - 어디에 신청 : 서울시
 - 어떻게 신청 : 온라인 신청

2. 강북구 청년네트워크 운영

- ◆사업 개요
 - 사업 기간 : 상시 운영
 - 목적 : 청년참여기구를 통한 강북구 청년정책 의제 발굴 및 청년활동 추진
- ◆신청 자격(핵심 요건)
 - 연령 : 19세 ~ 39세
 - 주거 : 서울특별시 강북구
 - 소득 : 제한 없음
 - 기타 조건 : 청년 네트워크 멤버십 혜택 제공
- ◆지원 금액·기간
 - 월 지원 금액 : 없음
 - 지원 기간 : 상시 운영
- ◆신청 방법(절차)
 - 어디에 신청 : 강북구 청년네트워크 신청 링크
 - 어떻게 신청 : 온라인 신청

3. 대학생 동아리 사회기여활동 지원

- ◆사업 개요
 - 사업 기간 : 2025. 1.~12.
 - 목적 : 대학생 동아리 지원을 통해 사회적 관계 회복과 자발적인 연대 및 협력 경험 제공
- ◆신청 자격(핵심 요건)
 - 연령 : 대학생
 - 주거 : 서울특별시
 - 소득 : 제한 없음
 - 기타 조건 : 서울 소재 대학교 소속 동아리

➤ Case 2 전국

"전국 단기집단상담프로그램" -> 전국대상 정책만 포함

질문: 전국 단기집단상담프로그램

■전체 답변:

안녕! 나는 청년들의 든든한 정책 선배, 청년이름 선배봇 🐶이야.
주거, 월세, 일자리, 복지 정책 등 궁금한 점이 있으면 언제든지 나에게 물어봐! 🐶

사용자 질문 : 전국 단기집단상담프로그램

답변 :

1. 단기집단상담프로그램

- ◆사업 개요
 - 사업 기간 : 연중 운영
 - 목적 : 구직자가 취업의욕을 고취하고 직업기초능력을 향상시키기 위한 체험·실습식 프로그램
- ◆신청 자격(핵심 요건)
 - 연령 : 제한 없음
 - 주거 : 제한 없음
 - 소득 : 제한 없음
 - 기타 조건 : 구직자
- ◆지원 금액·기간
 - 월 지원 금액 : 해당 없음
 - 지원 기간 : 과정별 3시간
- ◆신청 방법(절차)
 - 어디에 신청 : 서울특별시 및 부산광역시의 해당 지역
 - 어떻게 신청 : 해당 지역의 상담센터에 문의하여 신청

2. 성취프로그램

- ◆사업 개요
 - 사업 기간 : 연중 운영
 - 목적 : 구직기술 향상을 통해 자신감 회복을 지원하는 집단상담 프로그램
- ◆신청 자격(핵심 요건)
 - 연령 : 제한 없음
 - 주거 : 제한 없음
 - 소득 : 제한 없음
 - 기타 조건 : 구직자
- ◆지원 금액·기간
 - 월 지원 금액 : 해당 없음
 - 지원 기간 : 4월 동안 총 24시간 운영 (1월 6시간)
- ◆신청 방법(절차)
 - 어디에 신청 : 서울특별시 및 부산광역시의 해당 지역
 - 어떻게 신청 : 해당 지역의 상담센터에 문의하여 신청

3. 고용평등 심층상담(직장 내 성희롱, 고용상 성차별 피해, 모성보호제도 등)

- ◆사업 개요
 - 사업 기간 : 연중 운영
 - 목적 : 직장 내 성희롱, 고용상 성차별, 모성보호제도 등 피해와 관련한 전문 상담 서비스 제공
- ◆신청 자격(핵심 요건)
 - 연령 : 제한 없음
 - 주거 : 제한 없음
 - 소득 : 제한 없음
 - 기타 조건 : 상담을 원하는 모든 사람
- ◆지원 금액·기간
 - 월 지원 금액 : 해당 없음
 - 지원 기간 : 상담 필요 시 언제든지
- ◆신청 방법(절차)
 - 어디에 신청 : 전국 권역별 고용평등 전담 상담창구
 - 어떻게 신청 : 전화상담(1551-9811), 대면상담, 온라인 상담(고용노동부 노동포털)

◆ 출처:

- 윤통청년 청년정책 포털 (<https://youthcenter.go.kr>)

✓요약:

1. 단기집단상담프로그램: 구직자를 위한 체험·실습식 프로그램으로, 연중 운영되며 신청 자격은 제한 없음. 지원 기간은 과정별 3시간
2. 성취프로그램: 구직기술 향상을 위한 집단상담 프로그램으로, 연중 운영되며 신청 자격은 제한 없음. 지원 기간은 4월 동안 총 24시간
3. 고용평등 심층상담: 직장 내 성희롱, 성차별, 모성보호제도 관련 전문 상담 서비스로, 연중 운영되며 신청 자격은 제한 없음. 상

➤ Case 3 지역 미명시

“단기집단상담프로그램” -> 전국대상 정책만 포함, Case2와 결과가 같음을 확인하였다.

■질제 답변:
안녕! 나는 청년들의 든든한 정책 선배, 청년이음 선배봇 🐶이야.
주거, 월세, 일자리, 복지 정책 등 궁금한 점이 있으면 언제든지 나에게 물어봐! 🐶

사용자 질문 : 단기집단상담프로그램

답변 :

1. 단기집단상담프로그램

- ◆사업 개요
 - 사업 기간 : 연중 운영
 - 목적 : 구직자가 취업의욕을 고취하고 직업기초능력을 향상시키기 위한 체험·실습식 프로그램
- ◆신청 자격(핵심 요건)
 - 연령 : 제한 없음
 - 주거 : 제한 없음
 - 소득 : 제한 없음
 - 기타 조건 : 구직자
- ◆지원 금액·기간
 - 월 지원 금액 : 없음
 - 지원 기간 : 과장별 3시간
- ◆신청 방법(절차)
 - 어디에 신청 : 서울특별시 및 부산광역시의 해당 지역
 - 어떻게 신청 : 해당 지역의 상담센터에 문의하여 신청

2. 성취프로그램

- ◆사업 개요
 - 사업 기간 : 연중 운영
 - 목적 : 구직기술 향상을 통해 자신감 회복을 지원하는 집단상담 프로그램
- ◆신청 자격(핵심 요건)
 - 연령 : 제한 없음
 - 주거 : 제한 없음
 - 소득 : 제한 없음
 - 기타 조건 : 구직자
- ◆지원 금액·기간
 - 월 지원 금액 : 없음
 - 지원 기간 : 4일 동안 총 24시간 운영 (1일 6시간)
- ◆신청 방법(절차)
 - 어디에 신청 : 서울특별시 및 부산광역시의 해당 지역
 - 어떻게 신청 : 해당 지역의 상담센터에 문의하여 신청

3. 고용평등 심층상담(직장 내 성희롱, 고용상 성차별 피해, 모성보호제도 등)

- ◆사업 개요
 - 사업 기간 : 연중 운영
 - 목적 : 직장 내 성희롱, 고용상 성차별, 모성보호제도 등 피해와 관련한 전문 상담 서비스 제공
- ◆신청 자격(핵심 요건)
 - 연령 : 제한 없음
 - 주거 : 제한 없음
 - 소득 : 제한 없음
 - 기타 조건 : 직장 내 성희롱, 고용상 성차별, 모성보호제도 등 피해를 입은 사람
- ◆지원 금액·기간
 - 월 지원 금액 : 없음
 - 지원 기간 : 상담 필요 시 연중 운영
- ◆신청 방법(절차)
 - 어디에 신청 : 전화상담(1551-9811), 대면상담(전국 권역별 고용평등 전담 상담창구), 온라인 상담(고용노동부 노동포털)
 - 어떻게 신청 : 전화 또는 온라인을 통해 상담 신청

◆ 출처:

- 윤통청년 청년정책 포털 (<https://youthcenter.go.kr>)

✓요약:

1. 단기집단상담프로그램: 구직자를 위한 체험·실습식 프로그램으로, 연중 운영되며 신청 자격은 제한 없음. 지원 금액은 없고, 과장별

2. 성취프로그램: 구직기술 향상을 위한 집단상담 프로그램으로, 연중 운영되며 신청 자격은 제한 없음. 지원 금액은 없고, 4일 동안

3. 고용평등 심층상담: 직장 내 성희롱, 고용상 성차별, 모성보호제도 관련 전문 상담 서비스로, 연중 운영되며 신청 자격은 제한 없음

3.2.2 Multi-Query 확장성 테스트

단일 쿼리 대비 확장 쿼리가 다양한 키워드를 포함하는지 확인한다.

- Input: "취업 면접 수당에 대해 알고 싶습니다."
- Expected Queries: "청년 구직 활동 지원금", "면접비 지원 사업", "취업 역량 강화 프로그램" 등 유의어 포함 여부

MultiQueryGenerator 결과

=====

원본 쿼리: 취업 면접 수당에 대해 알고 싶습니다.

LLM 생성 결과:

전국 취업 면접 수당 정책

청년 구직 활동 지원금, 면접비 지원 사업, 취업 역량 강화 프로그램

면접비 지원, 구직자 면접 수당, 취업 지원금 관련 정책

최종 쿼리 (4개):

1. 취업 면접 수당에 대해 알고 싶습니다.
2. 전국 취업 면접 수당 정책
3. 청년 구직 활동 지원금, 면접비 지원 사업, 취업 역량 강화 프로그램
4. 면접비 지원, 구직자 면접 수당, 취업 지원금 관련 정책

3.2.3 응답 품질 테스트

최종 답변이 문서에 기반하여 환각 없이 작성되었는지 확인한다.

확인사항:

- ✓ 없는 내용을 지어내지 않았는가? (Hallucination)
- ✓ 출처를 명시했는가?
- ✓ 친근한 어조(페르소나)를 유지했는가?

4. 문제점 및 해결과정

4.1 문제 1: 지역을 바꿔서 질문해도 서울·전국 정책이 섞여서 나옴

4.1.1 문제 정의

쿼리에서 지역 정보를 받았지만, 실제 검색 필터에서 지역을 거르지 못해서 전국 정책 + 특정 지역 정책이 한 번에 섞여 반환되었다.

4.1.2 원인

벡터 검색 결과에서 `metadata["region"]` 값을 검사해 사용자 지역과 일치하는 정책만 남기도록 필터 로직 추가하였지만, 전처리된 데이터 자체에 지역 정보를 명시하지 않았다.

4.1.3 해결

추후 여러 테스트 과정을 거치면서 'zipCD' 필드에서 지역 정보를 가져와 '지역' 필드를 추가하고 '지역범위'까지 추가하였다.

4.2 문제 2: Retriever 결과가 없을 때, 전체 파이프라인이 바로 종료됨

4.2.1 문제 정의

검색 Retriever 결과가 없을 때, 전체 파이프라인이 바로 에러를 발생하고 종료되었다.

4.2.2 원인

검색 결과 리스트가 비어 있는데도, 이후 단계에서 `docs[0]`처럼 인덱싱을 시도해서 `IndexError`가 발생하였다.

4.2.3 해결

검색 결과가 0개일 경우 "관련 정책을 찾지 못했다"는 안전한 답변을 보내거나 기본 안내 문구를 반환하도록 가드 로직 추가하였다.

4.3 문제 3: 같은 구 이름을 검색했을 때, 다른 시/도의 정책까지 섞여서 나옴

4.3.1 문제 정의

'중구'와 같은 구는 여러 시/도에 있기 때문에 원하는 시/도의 정책이 나오지 않았다.

4.3.2 원인

파이프라인에서 "사용자가 말한 시/도" 와 "문서 metadata의 시/도"를 비교·검증하는 단계가 없었다.

4.3.3 해결

사용자가 말한 시/도 기준으로 한 번 더 필터링하는 로직을 파이프라인에 추가하였다

4.4 문제 4: ChromaDB에서 지역이 필터링되지 않음

4.4.1 문제 정의

SelfQueryRetriever를 통해 메타데이터를 필터링하려고 하였다. '대구'검색 시 벡터 유사도가 높은 다른 지역이 상위에 뜨는 문제가 발생하였다.

4.4.2 원인

ChromaDB는 복잡한 논리 연산자를 효과적으로 지원하지 않아, 우리가 원하는 (특정 지역) OR (전국) 조건을 DB에서 완벽히 거를 수 없었다.

4.4.3 해결

SelfQueryRetriever를 제거하고 RegionFilter 클래스를 구현하였다.

검색 전(Pre-Filtering) 단계에서는 전국 정책 여부만 판단하고, 사용자가 특정 지역(예: '대구')을 입력한 경우 DB 필터링 없이 전체 문서를 대상으로 검색하였다.

이후 검색 후(Post-Filtering) 단계에서 저장된 지역 정보('대구')를 기준으로 '대구 정책'과 '전국 정책'만 선별하도록 로직을 분리해 정확도를 확보하였다.

5. 결론 및 향후 과제

5.1 최종 평가

Query Routing, Multi-Query, Hybrid Search, Re-ranking 기법을 적용하여, 사용자의 거주 지역 정보를 기반으로 필터링을 수행하고 사용자가 해당 지역에서 실제로 받을 수 있는 정책만을 제시했다.

5.2 향후 과제

5.2.1 속도 향상

현재 Multi-Query(3개) × Ensemble(2개) 구조로 인해 최소 6회 이상의 검색이 실행된다. 이 과정에서 응답 지연이 발생하여 체감 속도가 다소 느리게 느껴질 수 있다. (약 5~7초)

향후 비동기 처리를 도입해 병렬 검색을 수행하여 전체 응답 시간을 단축하는 방안을 고려할 수 있다.

5.2.2 메타데이터 세분화

현재는 '지역' 중심의 필터링을 적용하고 있으나, 이에 더해 '연령', '소득분위', '취업상태' 등 추가 메타데이터를 반영하여 개인 맞춤형 정책 추천의 활용성을 더욱 강화할 수 있다.