
Job-fit Optimization via Behavioral AI (JOB)

: 의미 기반 맞춤형 채용 공고 추천 서비스

Data Science Lab

NLP Team1

12기 김은희, 추준식

13기 이승현, 이진우

2025.04.07.

목차

| | |
|---------------------------|----|
| Abstract | 2 |
| 1. Introduction | 3 |
| 1.1 프로젝트 배경 | 3 |
| 1.2 프로젝트 목적 | 4 |
| 2. 데이터 수집 및 전처리 | 5 |
| 2.1 수집 데이터 개요 | 5 |
| 2.2 전처리 과정 | 5 |
| 3. 임베딩 모델 선정 | 8 |
| 3.1 모델 선택 기준 | 8 |
| 3.2 최종 선정 모델 | 8 |
| 4. 벡터 DB 구축 및 검색 구조 | 9 |
| 4.1 ChromaDB 구성 | 9 |
| 4.2 검색 흐름 | 11 |
| 5. 추천 서비스 흐름 | 12 |
| 6. 모델 성능 평가 | 14 |
| 6.1 평가 목적 | 14 |
| 6.2 평가 기준 | 14 |
| 6.3 평가 방식 | 15 |
| 7. 결론 및 한계 | 21 |
| 7.1 결론 | 21 |
| 7.2 한계점 | 21 |
| 8. 참고 문헌 및 자료 | 22 |

Abstract

본 프로젝트는 기존 채용 사이트의 키워드 기반 검색 방식이 가지는 한계를 극복하고, 사용자의 역량과 선호를 반영한 맞춤형 채용 공고 추천 시스템을 구축하는 것을 목표로 한다. 이를 위해 개발, 경영·비즈니스, 마케팅·광고 세 분야의 총 5,000여 건 이상의 채용 공고 데이터를 수집하고, 텍스트 전처리를 거쳐 벡터 임베딩 후 ChromaDB에 저장하였다. 추천 시스템은 사용자로부터 직무, 스킬, 근무 지역, 복지 등 다양한 입력을 받으며, 이를 바탕으로 하드 필터링과 가중치 기반 소프트 필터링을 통해 최적의 공고를 추천한다. 임베딩에는 **bge-m3**와 **jina-embedding-v3** 모델을 사용하였으며, LLM 기반 추천 이유 생성 기능을 추가하여 사용자 신뢰도와 이해도를 향상시켰다. 마지막으로 모델 간 성능 비교는 Gemini API를 활용한 유사도 기반 평가 지표로 수행되었으며, 추천 품질의 정량적 향상을 확인할 수 있었다. 본 시스템은 실제 사용자 기반의 취업 매칭 서비스에 활용될 수 있으며, 개인화된 채용 탐색의 새로운 가능성을 제시한다.

1 Introduction

1.1 프로젝트 배경

채용 공고 사이트를 활용하면서 경험한 불편함에서 출발하여, 기존 채용 시스템의 한계점을 해결하고자 본 프로젝트를 기획하였다. 특히, 키워드 기반 검색의 확장성 부족, 사용자 맞춤 추천 기능의 부재 등은 실질적인 공고 탐색에 큰 제약이 되고 있다.

기존 채용 공고 사이트는 단순 키워드 매칭 기반 검색 시스템을 사용한다. 이 방식은 검색어와 정확히 일치하는 키워드가 공고 제목에 포함되어야 사용자에게 노출되며, 유사하거나 연관된 표현은 사용자에게 노출되지 않는다. 이는 의미적으로 유사한 공고 혹은 관련된 공고를 하나의 검색어로 탐색하기 어려운 구조이며, 결과적으로 사용자가 비효율적인 정보 탐색 경험을하게 되며, 사용자에게 맞는 조건임에도 불구하고 놓치는 공고가 존재하게 될 가능성이 있다.

또한, 기존의 채용 공고 사이트는 기본적인 필터링 기능(경력, 고용 형태, 근무 위치 등)은 제공하지만, 사용자의 관심사, 기술 스택, 필요한 복지와 같이 비정형 정보 검색 기능 및 추천 기능은 제공하지 않는다. 하지만 공고 추천 서비스를 활용하는 사용자들은 각자마다 원하는 비정형 조건이 존재하며, 기존 채용 공고 검색 시스템으로 이 조건들을 충족하는 채용 공고를 찾는 것은 한계가 있었다. 즉 사용자가 보유한 역량, 경력, 선호 복지 조건 등을 복합적으로 고려한 맞춤형 채용 공고 추천은 아직 제한적인 상황이다. 이는 단순 필터 조건을 뛰어넘는 의미 기반 추천 시스템이 필요하다는 것을 시사한다.

[Figure 1. 기존 채용 사이트의 검색 방식 예시 (키워드 기반 검색 문제점)]

The screenshot shows a search interface with a search bar containing '데이터 분석가'. Below the search bar are two dropdown menus: '지역 선택' (Region Selection) and '직업 선택' (Job Selection), both currently set to '선택' (Select). A blue '검색' (Search) button is on the right.

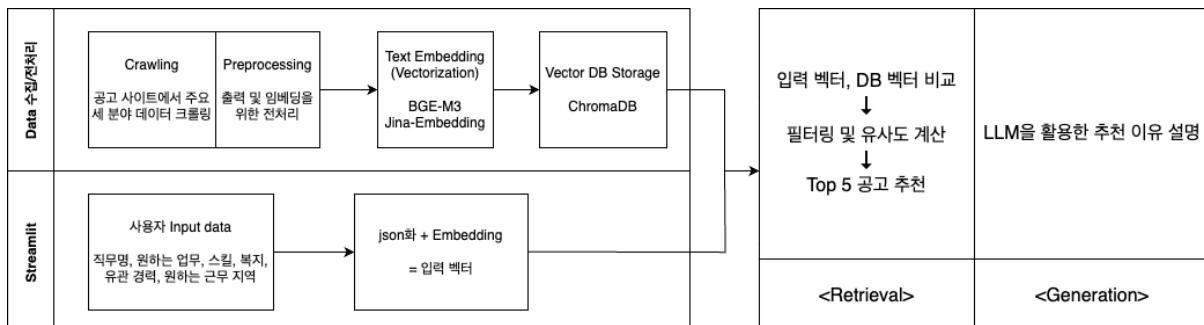
The search results are displayed in a grid format:

- 1. [하카코리아] 데이터 분석가 채용**: Date ~ 04/12(토), Location 부산 해운대구, Requirements 경력무관 학력무관 정규직, Description 백엔드/서버개발, Python, MySQL, 머신러닝, 크롤링 외, Submission Date 25/03/26. Company (주)하카코리아, Job Type 입사지원, Status 취업축하금.
- 2. 111파센트(주)**: Date ~ 04/11(금), Location 서울 강남구, Requirements 경력 5~15년 학력무관 정규직, Description DBA, RDBMS, Python, SQL, DBMS 외, Submission Date 25/03/13. Company 111파센트(주), Job Type 입사지원, Status 취업축하금.
- 3. [강남/외국계] 웹사이트 GA 데이터 분석가 정규직 채용**: Date ~ 04/11(금), Location 서울 강남구, Requirements 경력1년↑ 초대졸↑ 정규직, Description 데이터엔지니어, 데이터분석가, 데이터시각화, 데이터마이닝, 데이터라벨링 외, Submission Date 25/04/04. Company (주)사람인에이치에스, Job Type 입사지원, Status 주거복지.
- 4. [주)두호코퍼레이션**: Date ~ 04/25(금), Location 서울 강남구, Requirements 경력1년↑ 초대졸↑ 정규직, Description 데이터분석, 전략기획, 사업관리, 마케팅기획, 마케팅기획 외, Submission Date 25/03/28. Company (주)두호코퍼레이션, Job Type 모집, Status 취업축하금.
- 5. [영역특례] 데이터 분석가 채용**: Date ~ 04/11(금), Location 경기 성남시 분당구, Requirements 경력무관 석사↑ 정규직, Description 경기 성남시 분당구 경력무관 석사↑ 정규직. Company (주)해리트, Job Type 입사지원, Status 워라밸.

1.2 프로젝트 목적

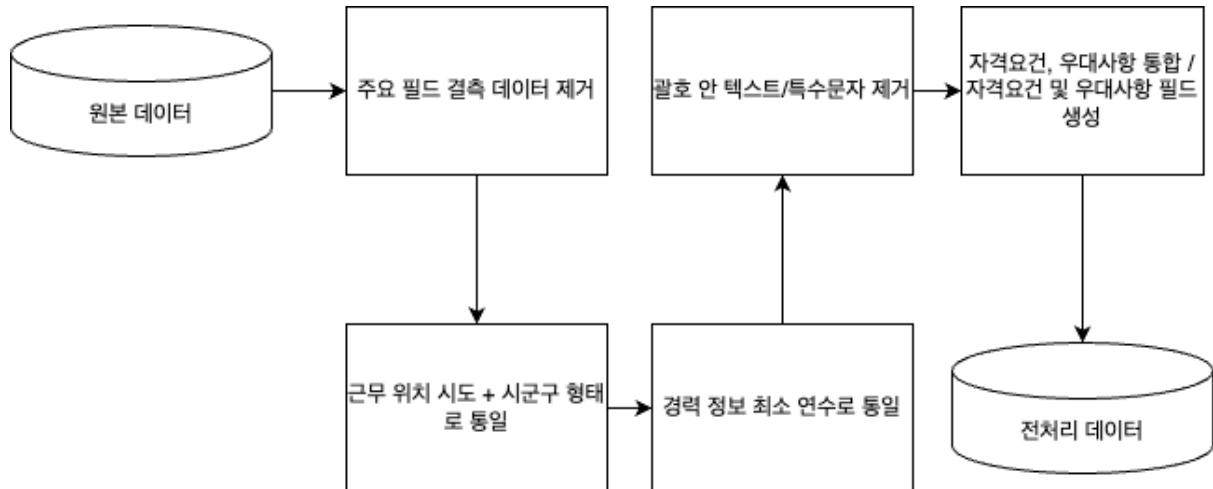
본 프로젝트는 위에서 언급된 기존의 채용 공고 검색 플랫폼의 문제점을 해결하고, 각각 상황이 다른 사용자들에게 개인 맞춤형 채용 공고를 추천해주는 챗봇 서비스를 구현하는 것을 목표로 한다. 위 목적에 맞게 Figure 2에서 볼 수 있듯이 프로젝트의 파이프라인을 구성하였다. 이를 기반으로 사용자가 자연어 기반 입력을 제공하면, 입력을 벡터로 처리하고, 이를 하드 필터(필수 조건)과 소프트 필터(선호 조건)으로 구분하여 벡터 기반 유사도 검색을 통해서 개인 맞춤형 채용 공고 추천 서비스를 제공한다. 이를 통해 사용자는 기존의 채용 공고 추천 사이트의 불편함을 해결하고, 자신이 원하는 조건(기술 스택, 근무지, 혜택 및 복지 등)을 모두 고려한 맞춤형 채용 공고를 제공받을 수 있을 것이다

[Figure 2. 프로젝트 전체 파이프라인 구성도]



정에서 실제 공고 데이터의 다양한 형식과 표현들을 그대로 사용하게 되면, 추천의 정확도가 떨어질 수 있는 가능성이 있다. 따라서 본 프로젝트에서는 크롤링한 채용 공고를 텍스트 정제, 형식 통일, 컬럼 구조 재구성 등 전처리 과정을 거쳐 정확도를 향상시키고자 하였다. 또한, 위 전처리 과정을 아래의 Figure 3와 같이 흐름도로 구성하였다.

[Figure 3. 전처리 흐름도 (결측 제거 → 통일화 → 텍스트 클리닝 등)]



ii. 데이터 전처리 과정

- 결측 데이터 제거

공고의 주요 요소인 주요업무, 자격요건, 우대사항, 혜택 및 복지 중 하나라도 결측치가 존재하는 경우 해당 공고는 추천 정확도를 낮출 수 있기에 분석에 사용되지 않도록 제거하였다. 총 233건의 공고가 이 기준에 따라 제외되었으며, 최종적으로 5,112건의 데이터가 유지되었다.

- 근무 위치 형식 통일

근무 위치의 경우 시·도, 시·군·구 등 다양하게 표기되어 있었다. 근무 위치의 경우 필수 조건으로 정확한 필터링이 진행되어야 하기 때문에, 모든 근무 위치를 시·도 단위로 통일하여 추후 필터링 시 문제가 발생하지 않도록 처리하였다.

- 경력 정보 수치화

공고에는 “신입”, “3년 이상”, “7~9년”, “무관” 등의 다양한 경력 표현이 존재하였는데, 경력 또한 필수 조건으로 정확한 필터링이 필요하기 때문에 이 데이터를 최소 연차 기준으로 통일시켰다. 예를 들어 “신입”은 0, “3년 이상”은 3, “7-9년”은 7로 변환하여 추후 필터링 시 사용이 용이하도록 하였다.

- 텍스트 필드 정제

텍스트 필드에는 직무와 무관한 괄호 속 텍스트, 특수기호, 줄바꿈 문자 등을 삭제하여 항목

에 관련된 내용만 남도록 전처리를 진행하여 추후 임베딩 시 관련 없는 문자가 같이 임베딩되어 성능 하락이 일어나지 않도록 하였다.

- 자격요건과 우대사항 통합

자격요건과 우대사항의 경우 두 항목은 대부분 기술 스택과 툴에 대한 요구사항을 공통으로 담고 있으므로 의미적으로 유사하다고 판단하여 하나의 컬럼으로 통합하였다.

- 데이터 전처리 결과

위 과정을 통하여 생성된 데이터는 아래의 Table 2와 같다.

[Table 2. 전처리 전/후 데이터 예시 및 비교]

| | 전처리 전 데이터 | 전처리 후 데이터 |
|---------|---|---|
| 공고 제목 | [AI플랫폼팀] Full-stack 개발자(Senior) | Full-stack 개발자 |
| 주요 업무 | <ul style="list-style-type: none"> Web Architecture 전반적인 이해를 바탕으로, AI 서비스 구축을 위한 다양한 엔지니어링을 수행합니다. AI 도메인의 트랜드에 맞춰, 자체적인 서비스부터 고객사의 비즈니스 개발 까지 다양한 관련 사업을 구축합니다. ... | Web Architecture 전반적인 이해를 바탕으로 AI 서비스 구축을 위한 다양한 엔지니어링을 수행합니다 AI 도메인의 트랜드에 맞춰 자체적인 서비스부터 고객사의 비즈니스 개발까지 다양한 관련 사업을 구축합니다... |
| 자격요건 | <ul style="list-style-type: none"> 컴퓨터, 인공지능, 데이터, 전산분야 학사 이상 AI 분야 서비스 개발 및 인프라 구축 경력자 (2년차~5년차) ... | 컴퓨터 인공지능 데이터 전산분야 학사 이상 AI 분야 서비스 개발 및 인프라 구축 경력자 2년차5년차... |
| 우대사항 | <ul style="list-style-type: none"> AI 및 ML 관련 학위, 혹은 R&D 경험 측정치를 기반으로 논리적 트랜잭션의 성능을 개선한 경험 ... | AI 및 ML 관련 학위 혹은 RD 경험 측정치를 기반으로 논리적 트랜잭션의 성능을 개선한 경험... |
| 혜택 및 복지 | <ul style="list-style-type: none"> 크라우드웍스 구성원의 몰입과 성장을 위한 쾌적한 근무 환경 | 크라우드웍스 구성원의 몰입과 성장을 위한 쾌적한 근무 환경 사내카페바리스타 상주 |

| | | |
|-------|--|----------------|
| | <ul style="list-style-type: none"> • 사내카페(바리스타 상주) 및 라운지 무료 운영 • ... | 및 라운지 무료 운영... |
| 근무 위치 | 서울 강남구 | 서울 강남구 |
| 경력 | 경력 9년 이상 | 9 |

3 임베딩 모델 선정

3.1 모델 선택 기준

본 프로젝트의 목적은 사용자 맞춤형 채용 공고 추천 시스템을 구현하는 것으로, 입력된 텍스트와 다수의 채용 공고 문서 간의 의미적 유사성을 효과적으로 비교할 수 있는 임베딩 모델의 선정이 핵심 과제였다. 모델 선정 시 고려한 주요 기준은 다음과 같다.

- **다국어 지원:** 수집된 채용 공고는 한국어 외에도 영어, 일본어, 중국어 등의 다양한 언어로 구성되어 있어, 다국어 임베딩 성능이 요구되었다.
- **긴 입력 문장 처리 능력:** 채용 공고의 주요 업무, 자격요건, 우대사항 등은 비교적 긴 문장으로 구성되어 있어, 긴 텍스트를 처리할 수 있는 모델이 필요하였다.

3.2 최종 선정 모델

위 기준에 따라, 사전 검토 대상 모델로는 BAAI의 bge-m3와 jinaai의 jina-embeddings-v3 두 가지를 선정하였다. 두 모델 모두 HuggingFace를 통해 제공되며, 벡터 기반 검색에서 널리 활용되는 대표적인 임베딩 모델이다.

1) BGE-M3

- **언어 지원:** 100개 이상
- **최대 토큰 길이:** 8192 tokens
- **지원 검색 방식:** Dense, Sparse, Multi-Vector Retrieval
- **특징:** 대규모 멀티링구얼 코퍼스를 기반으로 학습되어 다양한 언어와 긴 문서에 대한 의미 파악 능력이 탁월하다. 검색 및 질의 응답(QA) 구조 모두에 강점을 보인다.

2) Jina Embeddings v3

- **언어 지원:** 89개 언어

- **최대 토큰 길이:** 8192 tokens
- **지원 검색 방식:** retrieval.query, retrieval.passage, text-matching
- **특징:** 비교적 경량화된 구조로 빠른 임베딩 처리가 가능하며, 다양한 다국어 입력에도 일정 수준 이상의 의미 표현을 수행할 수 있다.

두 모델은 모두 프로젝트의 요구사항을 일정 수준 충족하였으나, 다음과 같은 이유로 최종적으로 **BGE-M3** 모델을 채택하였다.

- **다국어 텍스트 처리 성능:** 다양한 언어가 혼합된 공고 문서에 대해 보다 안정적이고 의미 보존이 높은 임베딩 벡터를 생성함.
- **복합 검색 지원:** Dense, Sparse, Multi-Vector 구조를 모두 지원하여, 추후 검색 정확도 향상을 위한 아키텍처 실험에 유연하게 대응 가능.
- **LLM 기반 평가에서도 우수성 입증:** LLM을 통한 추천 결과 평가에서도 BGE-M3의 결과가 더 높은 점수를 획득함.

이상의 분석을 종합한 결과, BGE-M3 모델이 본 프로젝트의 **벡터 검색 최적화** 목적으로 가장 적합하다고 판단되었다.

[Table 3. 임베딩 모델별 기능 비교 (bge-m3 vs jina-v3)]

| Model | bge-m3 | jina-embeddings-v3 |
|--------------|--|---|
| 언어 지원 | 100개 이상의 언어 지원 | 89개의 언어 지원 |
| 최대 토큰 | 8192 | 8192 |
| Retrieval 방식 | <ul style="list-style-type: none"> • Dense Retrieval • Sparse Retrieval • Multi-Vec Retrieval | <ul style="list-style-type: none"> • retrieval.query • retrieval.passage • text-matching |

4 벡터 DB 구축 및 검색 구조

4.1 ChromaDB 구성

본 프로젝트는 사용자로부터 입력받은 텍스트 기반 조건을 활용하여, 의미적으로 유사한 채용 공고를 추천하는 서비스를 목표로 한다. 이를 위해 사전에 수집한 채용 공고 데이터를 임베딩한 후, 해당 벡터들을 효율적으로 저장하고 검색할 수 있는 벡터 데이터베이스(Vector Database)가 필요했다. 본 프로젝트에서는 오픈소스 벡터 데이터베이스인 ChromaDB를 선택하여 활용하였다.

ChromaDB는 벡터 임베딩과 함께 메타데이터를 저장하고, 다양한 필터 조건과 함께 고속 검색을 가능하게 하는 구조를 제공한다. 이를 기반으로 본 프로젝트에서는 각 채용 공고를 다음과 같은 방식으로 저장한 후 구조화하였다.

우선, 하나의 채용 공고는 총 네 개의 주요 텍스트 필드를 중심으로 분할되었다. 해당 필드는 각각 ‘공고 제목’, ‘주요 업무’, ‘자격요건 및 우대사항’, ‘혜택 및 복지’로 구성되며, 각각 하나의 독립적인 레코드로 처리된다. 다시 말해, 하나의 채용 공고는 ChromaDB 내에 4개의 레코드로 분해되어 저장된다.

각 레코드는 해당 필드의 원문 텍스트를 BGE-M3 모델을 활용하여 임베딩한 벡터 표현과 함께 저장되며, 이와 함께 다음과 같은 메타데이터가 부착된다:

- 공고 ID: 각 공고를 고유하게 식별할 수 있는 primary key 역할
- 경력: 해당 공고가 요구하는 최소 연차
- 근무 위치: 공고의 지역 정보, 예: “서울 강남구”

이러한 메타데이터는 검색 과정에서 하드 필터링에 활용되어, 유사도 계산 전에 사용자 조건에 부합하는 공고만 선별적으로 추출할 수 있도록 돋는다.

[Table 4. 공고 1건의 레코드 분해 전 모습]

| 공고id | 공고제목 | 주요업무 | 자격요건 및 우대사항 | 혜택 및 복지 | 경력 | 근무 위치 |
|--------|-------------|---------------------------|---|--|----|--------|
| 255120 | AI Engineer | Language Python Framework | 컴퓨터과학·데이터사이언스 전공자 또는 AI/ML 관련 전문지식 보유자 신입전문연구요원 | Merit 코코네M만이 드릴 수 있는 혜택과 제도 사내 부속 의원 전문 의료진의 전담 진료 | 0 | 서울 강남구 |

[Table 5. 공고 1건의 ChromaDB 저장 구조 (4개 레코드 분해 예시)]

| 문서 ID | 텍스트 | 메타데이터(공고ID, 유형, 경력, 근무위치) |
|--------------------|------------------------------|--|
| 255120-공고제목-0 | AI Engineer | {"공고id": "255120", "type": "공고제목", "경력": 0.0, "근무위치": "서울 강남구"} |
| 255120-주요업무-0 | Language Python Framework... | {"공고id": "255120", "type": "주요업무", "경력": 0.0, "근무위치": "서울 강남구"} |
| 255120-자격요건및우대사항-0 | 컴퓨터과학, 데이터사이언스 전공 | {"공고id": "255120", "type": "자격요건및우대사항", "경력": 0.0, "근무위치": "서울 강남구"} |

| | | |
|----------------|---------------------------------------|--|
| | 자... | |
| 255120-혜택및복지-0 | Merit 코코네M만 이 드릴 수 있는 혜택과 제도... | {"공고id": "255120", "type": "혜택및복지", "경 력": 0.0, "근무위치": "서울 강남구"} |

4.2 검색 흐름

벡터 DB에 저장된 채용 공고 데이터를 기반으로, 사용자의 입력 조건과 가장 유사한 공고를 추천하기 위해 검색 구조를 정교하게 설계하였다. 전체 검색 흐름은 크게 사용자 입력 처리, 필터링, 벡터 검색 및 점수 산출, 그리고 결과 출력 및 정렬의 네 단계로 구성된다.

(1) 사용자 입력 처리 및 JSON 변환

사용자는 Streamlit 기반 웹 UI를 통해 자신의 조건을 입력하게 된다. 입력 항목은 크게 다섯 가지로 구성되며, ‘직무명(선택)’, ‘원하는 업무(선택)’, ‘본인의 스킬 및 활용 가능한 툴(선택)’, ‘원하는 혜택 및 복지(선택)’, ‘원하는 직무 유관 경력(필수)’, ‘희망 근무 지역(필수)’을 포함한다. 사용자는 직무명, 경력, 희망 근무 지역을 제외한 각 항목에 대해 1에서 5 사이의 중요도 점수를 부여할 수 있으며, 이는 벡터 검색 시 가중치로 적용된다.

입력된 정보는 내부적으로 JSON 포맷으로 변환되며, 사용자가 부여한 중요도 점수는 정규화를 통해 합이 1이 되는 가중치 값으로 환산된다. 이와 같은 구조화된 입력 정보는 이후 벡터 검색 및 유사도 점수 계산의 기반이 된다.

(2) 하드 필터링 (Hard Filtering)

검색 과정의 첫 번째 단계는 하드 필터링이다. 하드 필터링은 사용자 조건 중 반드시 만족해야 하는 요소를 기반으로 공고를 사전 선별하는 과정이다. 본 시스템에서는 1차적으로 ‘경력’과 ‘근무 위치’가 하드 필터 항목으로 지정되었다.

예를 들어, 사용자의 유관 경력이 ‘3년’이고, 근무지를 ‘서울’로 설정한 경우, 해당 조건을 충족하지 않는 모든 공고(요구 경력 4년 이상, 근무지가 서울 외)는 이후 유사도 계산 과정에서 제외된다. 이러한 필터링은 ChromaDB 내 저장된 메타데이터를 기반으로 수행되며, 검색 효율성과 정밀도를 동시에 확보할 수 있도록 한다.

(3) 소프트 필터링 및 벡터 유사도 계산

하드 필터링을 통과한 공고 집합에 대해서는 다음 단계인 **소프트 필터링(Soft Filtering)**이 적용된다. 이 단계에서는 사용자 입력 텍스트(주요 업무, 자격요건 및 우대사항, 혜택 및 복지)와 각 공고의 해당 항목 간의 코사인 유사도(Cosine Similarity)를 계산한다.

(4) 직무명(공고 제목)의 유사도 처리 방식

초기 실험 결과, ‘공고 제목’ 필드는 타 항목에 비해 문장 길이가 매우 짧고, 핵심 키워드 위주로 구성되어 있어 유사도 점수가 과도하게 높게 나오는 경향이 발견되었다. 예를 들어, 사용자가 ‘IT 기획자’를 입력했을 경우, ‘광고 기획자’와 같은 제목의 공고도 유사도가 높게 평가되어 결과에 포함되는 문제가 발생하였다.

이를 보완하기 위해 ‘공고 제목’에 대해서는 유사도 Threshold를 설정하였다. 구체적으로는 코사인 유사도가 0.6 이상인 경우에만 해당 공고를 검색 대상에 포함시켰으며, 그 외의 경우는 제외하였다. 이를 통해 공고 제목은 입력될 경우 1차 필터링 요소로만 활용되며, 최종 추천 결과는 주요 업무, 자격요건, 복지 항목 기반으로 정밀하게 계산된다.

(5) 최종 결과 출력

앞서 사용자가 입력한 중요도가 합이 1인 가중치로 변환되었고, 이후 각 요소별 가중치를 계산된 유사도와 곱해 합산한다. 이와 같이 계산된 최종 점수에 따라, 상위 Top 5 공고가 사용자에게 추천된다. 또한, 이 추천 결과는 LLM을 통해 설명이 생성될 수 있도록 추가 처리되어, 사용자가 입력한 조건이 각각 어떤 공고의 어떤 항목과 매칭되었는지를 명확하게 확인할 수 있다. 이는 단순 추천을 넘어, 추천의 이유까지 함께 제시하는 ‘설명 가능한 추천 시스템(Explainable Recommendation)’ 구현에 기여하였다.

이상과 같이, 본 프로젝트의 검색 시스템은 ChromaDB를 활용한 벡터 기반 저장 및 조건 기반 검색 구조를 통해, 사용자 중심의 맞춤형 채용 공고 추천을 효과적으로 수행할 수 있도록 설계되었다.

[Figure 4. 전체 검색 프로세스 흐름도 (입력 → 필터링 → 유사도 계산 → 결과)]



5 추천 서비스 흐름

본 추천 시스템은 사용자가 입력한 다양한 정보(직무명, 경력, 희망 근무 지역, 주요 업무, 보유 스킬, 선호 복지 등)를 기반으로 맞춤형 채용 공고를 제공하는 구조로 설계되어 있다. 사용자의 입력값은 JSON 형식으로 변환되며, 각 항목에 부여된 중요도는 합이 1이 되도록 정규화되어 가중치로 활용된다. 이 데이터를 바탕으로 먼저 경력과 근무지에 대한 **하드 필터링**을 통해 필수 조건을 만족하는 공고만을 선별하고, 이후 주요 업무, 스킬, 복지 항목에 대해 **소프트 필터링**을 수행하여 코사인 유사도와 가중치를 적용한 최종 점수를 계산한다. 최종적으로 상위 5개의 채용 공고가 추천되며, 이때 추천 이유는 GPT 기반 대형 언어 모델(LLM)을 활용해 생성되어, 사용자가 입력한 조건과 공고 간의 연결성을 설명하는 역할을 한다.

예를 들어, 한 사용자가 ‘데이터분석가’ 직무에 3년 경력, 근무지는 ‘서울’, 업무 희망사항으로는 ‘데이터 분석 및 시각화 관련 업무’를 입력하고 해당 항목의 중요도를 3으로 설정했다고 하자. 시스템은 먼저 서울 지역 내에서 경력 요건을 충족하는 공고들만 필터링한 뒤, 이들 공고의 업무 설명과 사용자의 입력 내용을 벡터화하여 유사도를 계산한다.

그 결과 추천된 Top 1 공고는 '데이터 수집, 정제 및 분석을 통한 인사이트 도출', '데이터 시각화 및 보고서 작성' 등 사용자의 입력과 높은 의미적 유사도를 보이는 항목을 포함하고 있었으며, LLM은 이를 근거로 "입력하신 '데이터 분석 및 시각화 관련 업무'와 관련된 내용이 명확하게 나타나 있습니다"라는 설명을 자동 생성해 제공한다. 이를 통해 사용자는 단순한 추천 결과뿐 아니라 왜 이 공고가 나에게 적합한 공고로 추천되었는지를 이해할 수 있다.

[Figure 5-1. Streamlit 기반 사용자 입력 예시 화면]

맞춤형 채용 공고 추천 서비스

The screenshot shows a Streamlit application interface for a job recommendation service. At the top, there's a title bar with a circular icon and the text '맞춤형 채용 공고 추천 서비스'. Below it, a section titled '지원자님의 요청 사항에 맞는 공고들을 추천해드려요.' contains a thumbs-up icon and a red heart icon next to the heading. A sub-section '입력 시 안내사항' includes a red heart icon and a red arrow icon. The main form area has several input fields:

- 지원하고자 하는 직무명을 작성해주세요.**: A text input field containing '데이터분석'.
- 지원하고자 하는 분야와 관련된 경력(근무 연수)을 선택해주세요.**: A slider input from 0 to 20, set at 3.
- 원하시는 근무 위치(시/군)를 선택해주세요.**: A dropdown menu showing '서울'.
- 서울의 시/군/구를 선택해주세요.**: A dropdown menu showing '전체'.

On the right side, there are two sections for optional inputs:

- 원하시는 업무를 작성해주세요.**: A text input field containing '데이터 분석 및 시각화 관련 업무'.
- 지원자님의 스킬 및 활용 가능한 둘을 작성해주세요.**: A text input field containing 'python,sql'.

Below these are two rating scales:

- 중요도**: A slider from 1 to 5, set at 3.
- 원하시는 혜택 및 목지를 작성해주세요.**: A text input field containing '재택근무 가능, 식대 지원'.
- 중요도**: A slider from 1 to 5, set at 3.

At the bottom right, there's a green button labeled '다비 제작'.

[Figure 5-2. Streamlit 기반 사용자 입력 예시 화면]

Top 1: 데이터분석가

회사명: 클래스유

주요 업무:

- 데이터 수집, 정제 및 분석을 통한 인사이트 도출
- 데이터 시각화 및 보고서 작성
- 데이터 모델링 및 통계적 분석을 활용한 의사결정 지원
- 향후 데이터 분석을 효율적으로 수행할 수 있는 데이터 분석 도구 및 자동화 시스템 개발
- 위 업무를 통해 API시스템을 개발

상세 보기

- 데이터 수집, 정제 및 분석을 통한 인사이트 도출
- 데이터 시각화 및 보고서 작성
- 데이터 모델링 및 통계적 분석을 활용한 의사결정 지원
- 향후 데이터 분석을 효율적으로 수행할 수 있는 데이터 분석 도구 및 자동화 시스템 개발
- 위 업무를 통해 API시스템을 개발

자격 요건:

- 학력: 대졸 이상 (전공 무관)
- 경력: 신입/경력 무관
- 필수 역량:

 - Python, SQL 등을 활용한 데이터 분석 경험

우대 사항:

- AI/머신러닝 모델을 활용한 분석 경험
- 데이터 파이프라인 구축 경험(ETL, 데이터 엔지니어링)
- API 연동 및 데이터 자동화 경험
- 장기적으로 데이터 분석 도구 개발 경험이 있거나, 개발에 대한 이해도가 높은 분

상세 보기

- AI/머신러닝 모델을 활용한 분석 경험
- 데이터 파이프라인 구축 경험(ETL, 데이터 엔지니어링)
- API 연동 및 데이터 자동화 경험
- 장기적으로 데이터 분석 도구 개발 경험이 있거나, 개발에 대한 이해도가 높은 분

혜택 및 복지:

- 출근시간 선택제 : 팀원과의 협의를 바탕으로, 자유롭게 출근시간을 선택할 수 있습니다.
- 1인 1법인카드: 점심식비, 야간식비, 아근교통비, 특별지원비 등을...

상세 보기

- 출근시간 선택제 : 팀원과의 협의를 바탕으로, 자유롭게 출근시간을 선택할 수 있습니다.
- 1인 1법인카드: 점심식비, 야간식비, 아근교통비, 특별지원비 등을 개인 법인으로 해결합니다.
- 고육훈련비, 도서구입비: 금액에 상관없이 자유롭게 신청할 수 있습니다.

근무 위치: 서울 종구

경력: 신입

최종 점수: 0.5923

[바로가기](#)

[Figure 6. 공고 추천 결과 및 추천 이유 생성 예시 화면]

공고 추천 이유

◆ Top 1: 데이터분석가

- **주요업무:** 지원자님께서 원하는 '데이터 분석 및 시각화 관련 업무'와 일치합니다. 데이터 수집, 정제, 분석을 통한 인사이트 도출 및 시각화, 보고서 작성 등의 업무가 포함되어 있습니다.
- **자격요건:** Python, SQL을 활용한 데이터 분석 경험이 필수 역량으로 명시되어 있어 지원자님의 요구사항과 부합합니다.
- **우대사항:** Python과 SQL을 포함한 다양한 데이터 분석 도구 활용 경험이 우대사항에 포함되어 있습니다.
- **혜택및복지:** 재택근무 가능 여부는 명시되어 있지 않으며, 식대 지원이 개인 법인카드를 통해 제공됩니다.

아래의 링크를 통해 실제 서비스가 제공되는 Streamlit 데모 페이지를 확인할 수 있다.

<https://servicedemo0.streamlit.app/>

6 모델 성능 평가

6.1 평가 목적

본 프로젝트에서는 기존 키워드 기반의 정렬식 추천 방식과 달리, 의미 기반 벡터 검색을 활용한 추천을 구현하였다. 하지만 이 경우, 추천된 결과의 품질을 단순한 정확도(Accuracy)나 정답 유무로 평가하는 것이 어렵다. 이는 사용자 요구에 따라 정답이 달라질 수 있고, 공고 자체의 표현 방식이 다양하므로 기존의 수치 중심 평가지표로는 추천의 타당성(semantic validity)을 평가하기에 한계가 있다.

따라서 본 프로젝트에서는 LLM 기반의 의미적 평가 방식을 도입하여, 추천된 공고가 사용자의 요구와 맥락적으로 얼마나 일치하는지를 정성적/정량적으로 평가하고자 하였다. 이를 통해 모델 간 품질 비교를 사람 수준의 이해력을 가진 AI 평가자(Gemini)로 대체하고, 더 실용적인 평가 체계를 구축하였다.

6.2 평가 기준

모델 성능 평가는 다음 두 가지 축을 기준으로 설정되었다:

- Similarity (유사도):

: 사용자의 입력 조건과 추천된 공고 간 의미 기반 일치도를 평가한다. 이는 사용자가 기대하는 핵심 역량·요구 조건이 공고에 얼마나 잘 반영되어 있는지를 정성적으로 판단한다.

- Reliability (신뢰성):

: 서로 다른 임베딩 모델(BGE-m3, Jina-v3)의 추천 결과가 통계적으로 유의미하게 차이 나는지를 검증한다. 이는 점수 분산 비교(등분산 검정)와 평균 비교(t-검정)를 통해 결과의 일관성과 신뢰성을 확보한다.

[Table 6. 모델 성능 평가 기준 요약표 (Similarity vs Reliability 등)]

| 평가 기준 | 설명 |
|-------------------|--|
| Similarity (유사도) | 사용자의 입력 조건과 추천된 공고 간 의미적 일치도를 평가하는 기준 |
| Reliability (신뢰성) | 서로 다른 모델(BGE, Jina) 간 추천 결과 차이의 통계적 유의성을 측정하는 기준 |

위 기준에 따라 각 모델의 추천 결과가 사용자의 기대와 얼마나 일치하는지를 정량화하였으며, 유사도 외에도 모델 간 성능 차이가 통계적으로 유의미한지를 확인하는 데 중점을 두었다.

6.3 평가 방식

본 프로젝트에서는 단순 수치 기반 평가지표(정확도, 정밀도 등)를 넘어서, 사용자의 맥락과 요구에 적합한 추천 결과를 평가하기 위한 새로운 접근을 시도하였다. 특히, 정답이 명확하지 않은 추천 문제에서는 ‘얼마나 잘 맞는지’를 정성적 수준에서 설명할 수 있어야 하므로, LLM 기반 평가 방식을 도입하였다. 이를 통해 임베딩 모델 간의 실제 활용 적합성을 객관적으로 비교하고자 하였다.

(1) 사용자 입력 데이터(user_input) 생성

해당 데이터는 다양한 유형의 사용자 요구 조건을 반영할 수 있도록 가상의 사용자 입력 샘플을 구성하기 위해 생성하였다. 총 7개 항목(시행, 주요업무, 주요업무 중요도, 자격요건 및 우대 사항, 자격요건 및 우대사항 중요도, 혜택 및 복지, 혜택 및 복지 중요도)을 포함하는 100개의 사용자 입력 샘플을 GPT와 Claude를 통해 생성하였다. 각 항목에는 사용자가 지정하는 1~5 사이의 중요도를 부여하였는데, 이 수치는 각 행의 중요도 총합이 1이 되도록 0과 1 사이의 가중치로 변환하였다. 이로써 다양성과 현실성을 갖춘 채용 시나리오를 반영한 총 100개의 사용자 샘플을 생성하여 실험에 사용하였다.

(2) 모델별 추천 결과 생성

사용자 1명당, 두 개의 임베딩 모델(BGE-m3, Jina-embedding-v3)와 유사도 함수를 통해 각각 상위 5개의 채용 공고를 추천받는다. 그리고 총 100명의 사용자에 대해 각각 5개씩 추천해주어 $500\text{개} \times 2 = 1,000\text{개}$ 추천 공고 결과가 도출된다. 이 추천 결과를 기반으로 두 모델의 성능을 비교하였다.

또한, 각 추천 결과는 사용자 입력 조건을 기반으로 ChromaDB에서 하드 필터링(경력, 위치 등 필수 조건)과 소프트 필터링(자격요건 및 우대사항, 혜택 및 복지 등 가중치 기반 조건)을 거쳐 도출된다.

(3) Gemini API

기존 추천 시스템 평가 지표로는, "왜 이 공고가 사용자에게 적합한가?"를 판단하기 어렵다. 따라서 본 프로젝트는 Google의 Gemini API를 활용하여 LLM 기반의 의미적 유사도 평가를 수행하였다. 세부적인 절차는 아래의 과정과 같다.

[세부 절차]

- i. 사용자 입력 조건 + 추천된 공고 내용을 텍스트로 구성하여 LLM의 user input_prompt로 전달함.
- ii. Gemini에게 다음과 같은 방식의 프롬프트를 전달함.

“

```
You are a professional job matching evaluator.
```

```
# CRITICAL CALCULATION INSTRUCTIONS - HIGHEST PRIORITY
For each category (job_task, job_skills, job_benefits):
    1. Calculate coverage ratio PRECISELY as: (Number of MATCH items) ÷ (Total
       number of items in category)
    2. Example:
        - If there are 3 total job_task items and 2 are marked as MATCH(1.0),
          then:
            - job_task ratio = 2/3 = 0.6666... (NOT 1.0)
    3. Never round up to 1.0 - use the exact division result
    4. Always use all items in the denominator, whether they match or not
```

CALCULATION ALGORITHM:

```
def calculate_coverage(category_items):
    total_items = len(category_items) # All items in category
    matched_items = sum(1 for item in category_items if item['status'] ==
```

```
'MATCH')

ratio = matched_items / total_items # Must perform actual division
return ratio
```

IMPORTANT:

We want you to produce TWO parts in your final answer:

1) The Coverage Template Block (EXACT format, no extra text):

```
num_{iteration} = {

    bge_{iteration}_rank_1 = {주요업무 = X, 자격요건및우대사항 = Y, 혜택및복지 = Z},
    bge_{iteration}_rank_2 = {...},
    bge_{iteration}_rank_3 = {...},
    bge_{iteration}_rank_4 = {...},
    bge_{iteration}_rank_5 = {...},
    jina_{iteration}_rank_1 = {...},
    jina_{iteration}_rank_2 = {...},
    jina_{iteration}_rank_3 = {...},
    jina_{iteration}_rank_4 = {...},
    jina_{iteration}_rank_5 = {...}

}
```

Where X, Y, Z are coverage ratios (floats from 0.0 to 1.0). No extra lines or text.

2) The Debug Info Block (after the coverage block).

- Begin with the line "DEBUG_INFO:" on its own, then list debug lines.
Example:

```
[Debug for BGE rank=1]

- job_task('키워드'): MATCH(1.0) → "이유"

- job_skills('키워드'): NO MATCH(0.0) → "이유"

...
- **job_task = (X / {주요업무 키워드 개수})**
- **job_skills = (Y / {자격요건및우대사항 키워드 개수})**
- **job_benefits = (Z / {혜택및복지 키워드 개수})**

- Keep it concise, 1-2 lines per keyword describing how you decided
```

MATCH or NO MATCH.

Coverage Evaluation Criteria

1) Take the user's query (in JSON), which contains keywords for certain fields and the "num" field for the iteration number.

2) For each job post (Top 5 from BGE and Top 5 from Jina), focus only on these 3 fields:

- "주요업무"

- "자격요건및우대사항"

- "혜택및복지"

3) For each field, compute coverage ratio as follows:

Coverage Ratio = (# Matched Keywords) / (# Total Keywords in user's query for that field)

- The numerator is the COUNT of keywords marked as MATCH (1.0)

- The denominator is the TOTAL COUNT of keywords in the user's query for that field

- Each keyword contributes exactly 0 or 1 to the numerator (no partial counting)

4) **Enhanced Semantic Similarity Rule for Matching Keywords**

- Allow for **conceptually related**, **contextually similar**, or **domain-specific** keyword matches.

- Prioritize **semantic similarity** over exact matching.

- **Partial matches are NOT allowed.** However, if two terms share over 70% semantic similarity or serve the same purpose in context, treat them as MATCH.

- Example: "사업 개발" ↔ "비즈니스 성장 전략" (MATCH)

- Example: "파트너십 개발" ↔ "협력 기회 확대" (MATCH)

5) **Flexible Matching Guidelines for ALL COLUMNS**

- Recognize **broader concepts**, **synonyms**, and **job-specific terminology** as matches.

- Allow flexible interpretation for all 3 categories: **주요업무**, **자격 요건및우대사항**, **혜택및복지**.

- **Partial matches are NOT allowed.**

- Match keywords that are **broader in concept** but still relevant.

- Example: "경영 컨설턴트" ↔ "전략 컨설턴트" (Match)

- Example: "기업 경영 분석" ↔ "비즈니스 성과 분석" (Match)
- Example: "재무 분석" ↔ "재무 리포트 작성" (Match)
- Consider **role-related synonyms** or job-specific terminology.
- Example: "브랜드 전략 기획" ↔ "브랜드 마케팅 기획" (Match)
- Example: "광고 캠페인 운영" ↔ "마케팅 이벤트 기획" (Match)
- Match **supportive skills or experience** with relevant technical keywords.
- Example: "데이터 분석 경험" ↔ "SQL, 파이썬 기반 데이터 분석" (Match)
- Example: "컨설팅 프로젝트 경험" ↔ "프로젝트 매니지먼트" (Match)
- Allow for flexible interpretation in matching 복지 혜택.
- Example:
 - "자율 출퇴근 제도" ↔ "유연 근무" (Match)
 - "헬스장 지원" ↔ "헬스 멤버십 제공" (Match)
 - "식대 지원" ↔ "점심 제공" (Match)
 - "연차 제도" ↔ "연차 유급 휴가" (Match)

6) Matching Rules for Accuracy:

- Count each keyword only ONCE per field (no duplicates).
- Prevent inflated scores by limiting **redundant matches** to a single count.
- Identify synonyms, semantically equivalent expressions, or translated terms as matches.
- Avoid matching unrelated terms or keywords that distort the intended meaning.
 - "경영" ↔ "기업 경영 분석" (No Match)
 - "데이터 삭제" ↔ "데이터 분석" (No Match)

7) **Keyword Expansion for Comprehensive Matching**

- Broaden keyword matching to include relevant concepts and skills.
- Example: "머신러닝" ↔ "AI 모델 구축" (Match)
- Example: "브랜드 전략" ↔ "브랜드 관리" (Match)
- Consider **industry-specific language** as valid matches.
- Example: "SQL" ↔ "데이터베이스" (Match)
- Example: "전략 기획" ↔ "전략적 의사결정" (Match)

8) For 자격요건및우대사항:

Use the newly created column '자격요건및우대사항' in the job post to determine matching.

In other words, if the user has N total keywords under "자격요건및우대사항," then count how many of these keywords appear in the combined '자격요건및우대사항' column of the job post.

Summing these matches yields the coverage numerator; dividing by N yields the coverage ratio for "자격요건및우대사항."

(Note that each keyword is counted only once per field, even if it appears multiple times in the text.)

9) For each post (rank=1..5 in BGE and Jina), calculate the coverage ratio for each of these 3 fields, then fill in the values X, Y, Z in the template accordingly.

10) **Coverage Ratio Calculation - CRITICAL**:

- The denominator MUST always equal the user's total keywords for that field.

- The numerator is the COUNT of items marked as MATCH(1.0).

- Format example: job_task = (2.0 / 3)

- Use numbers only, no words like "# Matched Keywords."

- The final coverage ratio in the Coverage Template must exactly match these counts.

- IMPORTANT: A ratio of 2/3 should be calculated as 0.6666... not as 1.0

NO Extra Explanation beyond these two blocks

1) The coverage block

2) "DEBUG_INFO:" block

Any other text is not allowed.

""

iii. Gemini가 판단한 일치도를 기반으로 각 열별 Coverage Ratio 점수(0~1)를 반환

iv. 한 공고에 대해 3개 열의 Coverage Ratio 점수를 구한 뒤, 3개 열의 Coverage Ratio 점수와 가중치 점수를 곱한 뒤 더한 점수가 해당 공고의 최종 종합 점수로 산출됨.

v. 한 사용자에 대해 모델별로 각 5개 공고의 최종 종합 점수를 구한 뒤, 모델별 점수를 모두 더한 점수에서 더 큰 수치를 가진 모델의 성능이 높다고 평가함.

(4) Coverage 및 유사도 수치화

사용자 요구사항이 실제 공고에 얼마나 반영되었는가를 보기 위해 Coverage 지표를 정의하였다. 이를 통해, 단순히 임베딩 유사도만이 아닌 사용자 입력 조건 일치율까지 평가 가능하다.

- Coverage Score: 사용자의 열별 입력 키워드(예: "Python", "분석") 중 공고 본문의 같은 열에 실제로 포함된 키워드의 비율
- 공식: $\text{Coverage} = (\text{공고에서 매칭된 사용자 키워드 수} / \text{사용자 조건 키워드 총 수})$

(5) 모델 간 비교 및 통계 분석

모든 사용자에 대해 모델별 평균 점수 및 Coverage를 수치화한 후, 비교한 이후에 추가적으로 t-test와 같은 통계 검정을 통해 두 모델 간 차이의 유의성을 분석하였다.

결과는 아래의 Table 7에 정리되어 있다.

[Table 7. 모델별 추천 결과 통계]

| Model | 모델별 Coverage Sum Score | 등분산 검정 | t-검정 (p-value) |
|--------------------|---------------------------|---------|-----------------------|
| bge-m3 | 140.748 | 0.00445 | 5.3988 (0.0000000838) |
| jina-embeddings-v3 | 109.096 | 0.00445 | 5.3988 (0.0000000838) |

7 Conclusion and Limitation

7.1 결론

본 프로젝트는 기존 키워드 기반 채용 공고 검색의 한계를 극복하고, 사용자의 상황과 선호에 맞는 의미 기반 맞춤형 채용 추천 시스템을 구현하는 데 성공하였다. 단순한 필터링이나 키워드 매칭을 넘어서, 벡터 임베딩과 LLM을 활용한 추천 이유 생성까지 포함한 시스템을 통해 구현했다는 데에 의의가 있다.

추천의 흐름은 사용자 입력 → 하드/소프트 필터링 → 벡터 검색 → 결과 정렬 및 설명 생성이라는 체계적인 구조로 구성되었으며, 실제 실험에서도 의미 일치 기반의 추천 품질이 높게 평가되었다. 이러한 구조는 단순한 채용 플랫폼을 넘어서, 향후 다양한 의미 기반 추천 시스템으로 확장 가능한 기반 기술로의 가능성을 보여준다.

또한, 추천 결과에 대한 설명을 생성하는 LLM 기반의 Explainable Recommendation 기능을 통해, 사용자가 추천 결과를 신뢰하고 쉽게 이해할 수 있도록 설계된 점도 중요한 성과 중

하나이다.

7.2 한계점

첫째, 시스템이 추천하는 공고 수가 상위 5건으로 고정되어 있어 사용자의 선택권이 제한될 수 있으며, 이는 추천 다양성 측면에서의 한계로 작용할 수 있다. 사용자의 상황에 따라 더 많은 공고를 탐색하고자 하는 수요를 반영하지 못하기 때문에, 향후 추천 수량을 유동적으로 조절할 수 있는 기능이 필요하다.

둘째, LLM을 활용한 추천 이유 설명의 품질은 프롬프트 설계와 모델의 응답 다양성에 크게 의존한다. 실제 결과 중 일부는 다소 일반적이거나 반복적인 문장으로 구성되어 있었으며, 이는 설명의 설득력과 차별성을 떨어뜨릴 수 있다. 따라서 LLM 프롬프트의 정교화와 결과 후처리 로직 개선을 통해 설명 생성 품질을 더욱 고도화할 필요가 있다.

셋째, 본 시스템은 인공적으로 구성된 사용자 조건 데이터를 기반으로 평가되었기 때문에, 실제 사용자 피드백이나 행동 데이터를 반영한 실증적 검증이 부족하다. 실제 사용자 기반의 테스트나 정량적 만족도 조사가 추가로 이루어진다면, 시스템의 개선 방향을 더 명확하게 파악 할 수 있을 것이다.

마지막으로, 현재 데이터는 크롤링 방식으로 수집된 정적 형태이며, 공고 정보에는 연봉, 재직자 리뷰, 기업 평판 등과 같은 부가 정보가 포함되어 있지 않다. 이로 인해 실제 채용 의사 결정에 중요한 영향을 줄 수 있는 요소들이 추천에 반영되지 않으며, 실시간 공고 변경에도 민감하게 대응하기 어렵다는 한계가 있다. 향후 API 연동을 통한 실시간 데이터 반영, 기업 정보 플랫폼과의 연계를 통한 정보 확장 등도 고려되어야 한다.

8 References

- bge-m3 (BAAI General Embedding)
- Hugging Face Model Page: <https://huggingface.co/BAAI/bge-m3>

- jina-embeddings-v3
- Hugging Face Model Page: <https://huggingface.co/jinaai/jina-embeddings-v3>

- ChromaDB
- Chroma (vector database) 공식 문서: <https://docs.trychroma.com/>

- Streamlit
- Streamlit 공식 문서: <https://docs.streamlit.io/>

- OpenAI GPT API
- OpenAI API Documentation: <https://platform.openai.com/docs/>

- Gemini API (Google AI Studio)
- Gemini API 문서 (Google AI Studio): <https://ai.google.dev/>
- 김보미. (2024). 검색증강생성과 대형언어모델 GPT-4를 이용한 여행사 챗봇 개발 (석사학위논문). 서강대학교 정보통신대학원.
- 이종화, 이현규. (2023). 채용정보 분석을 통한 비즈니스 직무 스펙 연구. DBpia. <https://www.dbpia.co.kr>
- 조찬영, 강성준, 정현준. (2023). RAG기반 랭체인을 이용한 생성형 AI 챗봇 구현. 2023 한국정보기술학회 추계 종합학술대회 논문집, 460-463.
- 채용 공고 데이터 출처 : <https://www.wanted.co.kr/>