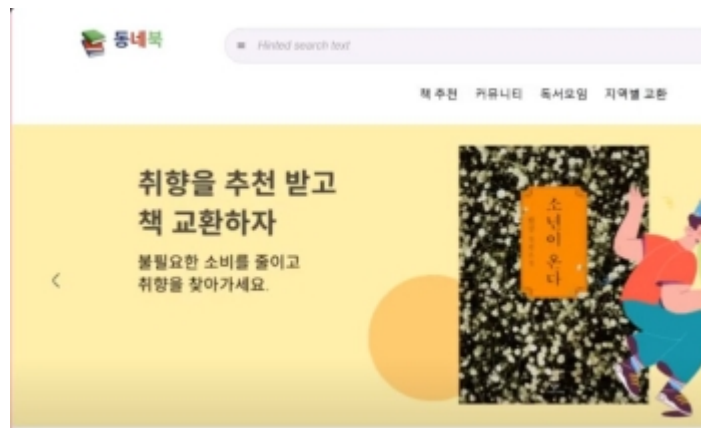


책 추천 시스템

동네북



1. 프로젝트 주제

지역 기반 도서 교환 커뮤니티를 위한 개인화 추천 및 키워드 시각화 서비스 '동네북' 개발

'동네북'은 지역 주민 간 도서 교환을 촉진하고, 사용자가 관심을 가질 만한 책을 효율적으로 탐색할 수 있도록 돕는 **책 추천 기반 플랫폼**입니다.

사용자 편의를 높이기 위해:

- 한국어 문맥 기반 **도서 추천 시스템(KLUE-BERT 기반)**
- 사용자의 읽은 책들에 대한 개인화 추천 알고리즘
- 도서 리뷰에서 도출된 핵심 키워드를 시각적으로 표현하는 **워드 클라우드 기능**
- 도서 등록 및 지역 기반 검색 기능

을 통합한 웹 기반 플랫폼을 구현하였습니다.

2. 프로젝트 제안 배경

사회적/서비스적 문제 인식

- 중고 도서의 가치 순환은 필요하지만, 단순 거래 플랫폼은 **도서의 내용이나 주제에 기반한 탐색 기능이 부족함**
- 독자는 "읽어볼 만한 책"을 찾기 어렵고, 관심 분야에 대한 책을 발견하기 위해 **직접 검색해야 하는 수고가 큼**

- 도서 추천 시스템이 있어도, 대부분은 **판매량/인기순 추천**에 치우쳐 있음

기술적 문제 인식

- 기존 추천 시스템은 단순한 키워드 매칭 기반으로, **문맥적 의미를 반영하지 못함**
- **한국어 데이터에 특화된 추천 시스템이 부족**, 영어 기반 모델을 그대로 사용하는 경우 성능 저하 발생
- 도서에 대한 주요 키워드를 사용자에게 제공하는 기능이 없거나, **가시적 정보가 결여**되어 있음

→ '동네북'은 **한국어 문맥을 반영한 추천 기능**과 **도서별 주요 키워드 시각화** 기능을 통해 **추천 탐색과 직관적 정보 이해**를 동시에 지원하고자 하였습니다.

3. 사용 개념 및 핵심 기술

- **자연어처리 (NLP)**: KLUE-BERT 기반 문맥 임베딩
- **추천 시스템**: 콘텐츠 기반 필터링, 코사인 유사도 계산
- **웹 크롤링**: 알라딘 리뷰 데이터 자동 수집
- **비지도 학습**: LDA 토픽 모델링 (도서 대표 키워드 추출 목적)
- **시각화**: WordCloud로 도서별 핵심 키워드 시각적 제공
- **서버 개발**: FastAPI + SQLAlchemy 기반 AI API
- **성능 최적화**: PyTorch + CUDA로 GPU 기반 임베딩 처리

4. 실험 계획 및 구성 전략

목표	기술 방식	설정 이유
도서 간 의미 기반 유사도 계산	메타데이터 임베딩 + 코사인 유사도 행렬 생성	문맥 정보를 반영한 추천 구조 확보
실시간 추천 응답 보장	사전 연산된 임베딩 및 유사도 캐싱 구조	시스템 부하 최소화 및 빠른 API 응답 시간 확보
도서에 대한 직관적 정보 제공	리뷰 텍스트 전처리 및 토큰화 + 키워드 추출 기법 활용	사용자에게 도서의 주요 주제나 분위기 전달
추천 결과의 다양성 유지	제목/시리즈 기반 유사도 필터링 적용	동일 도서, 시리즈 도서 중복 노출 방지로 추천 신뢰도 향상
사용자 맞춤형 추천	다중 도서 기반 유사도 벡터 합산 방식	사용자 선호도를 유사도 기반으로 종합하여 개인화 추천 구현

5. 실험 프로세스

1) 리뷰 데이터 수집 및 전처리

처리 방식

- 알라딘 웹사이트에서 도서의 ISBN13을 기반으로 리뷰 데이터를 직접 크롤링
- Python의 `requests` 및 `BeautifulSoup` 를 활용하여 자동화 스크립트 구현
- 약 64000개 도서의 모든 리뷰를 수집하여, 리뷰의 질적 다양성을 확보

전처리 과정

- HTML 태그 및 특수문자 제거
- `Okt` 형태소 분석기를 사용해 **명사** 단위로 토큰화
- 한 글자 단어, 불용어 제거 (`책`, `정말`, `이다`, `그런` 등)

설계 이유

- 리뷰 데이터는 알라딘이 실사용자 기반으로 보유한 공개된 리뷰로, **신뢰성과 도메인 일관성**이 확보됨
 - 키워드 기반 워드 클라우드를 만들기 위해 **명사 중심 분석**이 유리함
-

2) 도서 메타데이터 임베딩 및 유사도 행렬 생성

입력 구성

- 입력 필드: `title`, `author`, `publisher`, `category`, `description`
- 해당 필드는 `isbn13` 을 기반으로 호출한 API 응답에서 자동 추출
- 하나의 입력 문자열로 병합 후 **KLUE-BERT** 모델에 전달

임베딩 및 유사도 계산

- `[CLS]` 토큰의 벡터(768차원)를 추출하여 `.npy` 로 저장
- 모든 책 임베딩을 L2 정규화 후 **Cosine Similarity** 행렬 생성 (N x N)

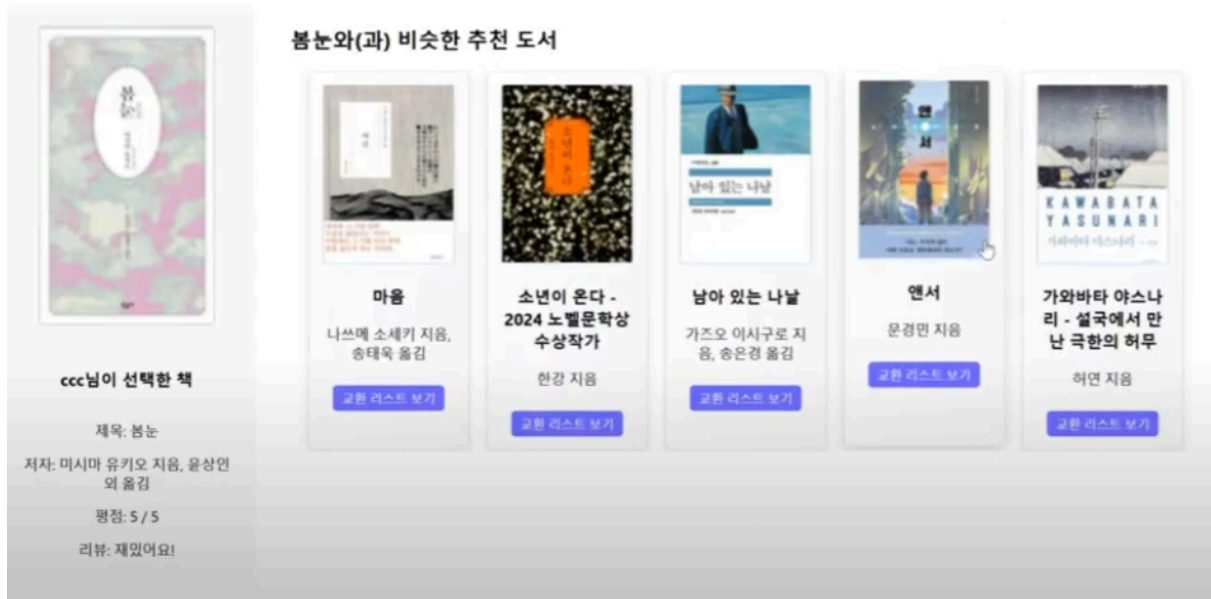
설계 이유

- KLUE-BERT는 한국어 문장 표현에 특화되어 도서 간 문맥 유사도를 효과적으로 추출
 - 사전 연산 및 `.npy` 캐싱 구조는 실시간 API 응답에 필요한 성능 확보에 효과적
-

3) 추천 알고리즘 설계

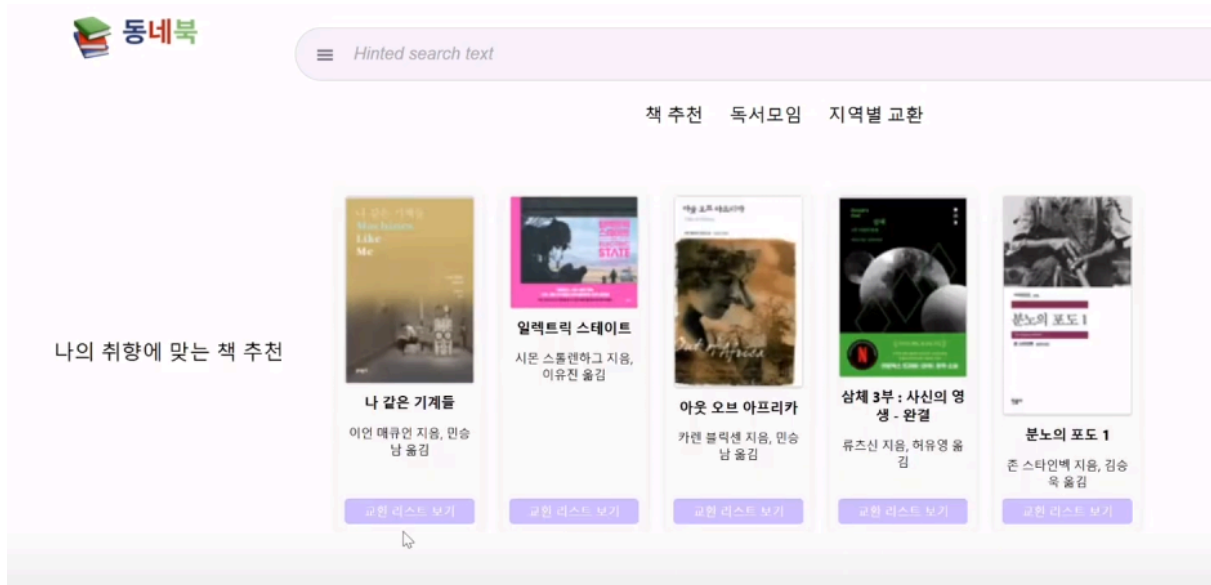
단일 도서 기반 추천

- 사용자가 선택한 책의 인덱스를 통해 유사도 행렬에서 해당 행 추출
- 유사도가 높은 순서대로 Top-N 도서 추출
- 제목 유사도 ≥ 0.8 , 동일 ISBN, 동일 시리즈 도서 제외



사용자 맞춤형 추천

- 사용자가 읽은 책 리스트의 인덱스를 기준으로 유사도 벡터 추출
- 각 유사도 벡터를 단순 합산하여 누적 유사도 계산
- Top-N 도서 선택 (이미 읽은 도서 제외)



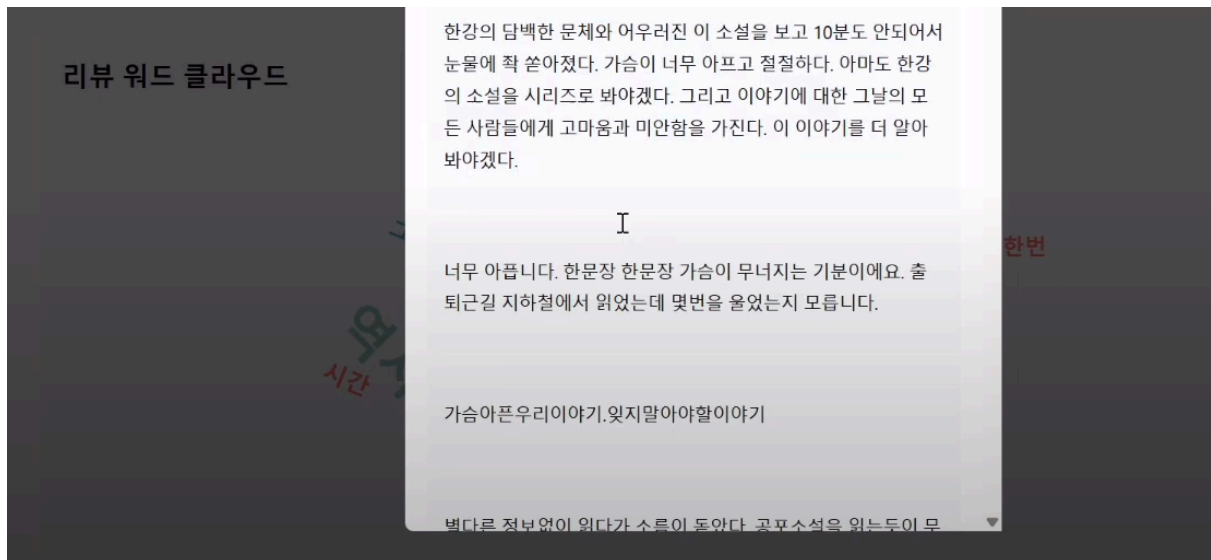
설계 이유

- 유사도 벡터 합산 방식은 사용자의 다양한 관심사를 반영하면서 연산 부담이 적고 직관적
- 벡터 평균 방식보다 실시간 응답 속도 및 연산량 측면에서 효율적

4) 리뷰 기반 키워드 시각화 (Word Cloud)

처리 방식

- 전처리된 리뷰 데이터를 기반으로 Gensim의 **LDA 모델** 학습
 - 토픽 수 K=5, 각 토픽당 키워드 5~10개 추출
- 빈도 기반이 아닌 **토픽 확률** 기반으로 키워드 가중치 설정
- Python **WordCloud** 라이브러리로 시각화 이미지 생성
- 워드 클라우드의 **키워드**를 클릭하면 키워드 포함리뷰 확인 가능



설계 이유

- 사용자가 책을 탐색할 때 **정서적·주제적 분위기를** 직관적으로 파악할 수 있도록 돕는 보조 도구
- 추천과는 독립적인 비지도 기반 정보 제공 기능

6. 실험 결과 및 학문적 고찰

6.1 추천 시스템 성능 평가

- 도서 메타데이터(`title` , `author` , `publisher` , `category` , `description`) 기반으로 생성된 문장 입력력을 **KLUE-BERT**에 전달하여 `[CLS]` 임베딩 추출
- **KoBERT, KoELECTRA, KLUE-BERT** 중 비교 실험 결과:
 - KLUE-BERT가 **추천 결과의 일관성, 문맥 반영 능력, 노이즈 내성 측면**에서 가장 우수
- 코사인 유사도 기반 유사도 행렬 구축 후:
 - **단일 도서 추천**: 정제된 유사도 기준으로 가장 유사한 책 Top-N 추천
 - **사용자 기반 추천**: 복수의 책 유사도 벡터 **단순 합산 방식** → 사용자 관심사 누적 반영
- 최종 시스템 응답 속도:
 - 사전 계산된 `.npy` 캐시 활용 → **평균 응답 시간 50ms 미만** 유지
 - 실시간 API 호출에도 무리 없는 퍼포먼스 확보

핵심 성과 요약

실험 항목	결과
추천 알고리즘 적합성	KLUE-BERT > KoBERT > KoELECTRA (비공식 평가 기준)
추천 응답 속도	평균 50ms 미만
사용자 기반 추천 구조	유사도 벡터 합산 방식 (단순하고 빠름)
캐싱 효율	NumPy 기반 전처리로 I/O 최소화

6.2 워드 클라우드 시각화 효과

- 알라딘에서 직접 크롤링한 사용자 리뷰 데이터를 바탕으로 **LDA 토픽 모델링** 수행
 - 각 도서당 리뷰가 충분한 경우, **도서 주제와 분위기를 나타내는 키워드** 도출 가능
 - 예: 『소년이 온다』 → **광주**, **죽음**, **운명**, **슬픔**, **저항** 등 키워드

6.3 한계점 및 기술적 고려 사항

- Cold Start 문제:**
신규 책의 경우 리뷰 부족 혹은 메타데이터 불충분으로 유사도 계산 및 키워드 도출이 제한됨
- 사용자 행동 로그 부재:**
추천 결과에 대한 정량적 지표(Precision@K, Recall@K 등) 평가 어려움
→ 협업 필터링 적용이나 사용자 기반 개선은 미실현 상태
- LDA의 모델 품질**은 리뷰 수와 텍스트 길이에 민감하며, 적은 수의 리뷰에서는 불안정한 결과가 나타남

6.4 참고 문헌 및 기술 기반

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova,

"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,"

Proc. NAACL-HLT, pp. 4171–4186, 2019.

<https://arxiv.org/abs/1810.04805>

[2] S. Park, H. Jeon, Y. Kang, and H. Shin,

"KLUE: Korean Language Understanding Evaluation,"

Proc. EMNLP, 2021.

<https://arxiv.org/abs/2105.09680>

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan,

"Latent Dirichlet Allocation,"

J. Mach. Learn. Res., vol. 3, pp. 993–1022, Jan. 2003.

<https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>

7. 향후 연구 및 확장 방향

- **프롬프트 기반 질의 추천 시스템 확장**

- 자연어 질의(예: "감동적인 현대문학") 입력 시 추천 결과 생성

- **사용자 로그 기반 강화학습 추천(RLHF)**

- 사용자 행동 로그(클릭, 찜 등)를 활용한 온라인 학습 구조 적용

- **도서 요약 및 리뷰 생성형 AI 연계**

- 추천 도서에 대해 자동 요약 또는 감성 리뷰 생성 기능 추가

- **LLM 한국어 벤치마크용 도서 데이터셋 구축**

- 메타데이터 + 리뷰 + 키워드를 통합한 도서 특화 NLP 학습용 벤치마크로 확장 가능