



# 개요

## [Project] LLM & RAG 기반 시나리오를 이용한 뷰티 제품 추천 시스템

### 1. 프로젝트 한 줄 요약

사용자의 구체적인 상황(시나리오) 텍스트를 입력받아, 11만 개의 아마존 상품 중 가장 적합한 제품을 찾아내고 추천 이유까지 설명해주는 검색형 추천 (Retrieval-Augmented Recommendation) 서비스

### 2. 문제 정의

- **기존의 한계:** 기존 추천 시스템(CF 등)은 “이거 산 사람이 저것도 샀어요” 식의 행동 패턴에만 의존하여, 사용자의 ‘현재 상황’이나 ‘구체적인 의도’를 반영하지 못함
- **니즈의 변화:** 사용자는 단순한 상품 나열이 아니라, “내 상황에 이 제품이 왜 필요한지”에 대한 납득할 만한 설명을 원함
- **해결책:** 사용자의 자연어 시나리오를 이해하는 의미 기반 검색 (Semantic Search) 기술과, 리뷰 데이터를 근거로 제시하는 LLM을 결합

### 3. 기술 아키텍처 (Technical Architecture)

#### Dataset

- **Amazon Reviews 2023** (<https://amazon-reviews-2023.github.io/>): User 632k / Item 112k / Rating 701k
- **Data Enrichment:** LLM을 활용해 Raw Review에서 “제품이 사용되는 상황/키워드”를 추출하여 아이템 메타데이터 강화

## Step 1: Retrieval (후보 추출)

- Model: Two-Tower Architecture (Bi-Encoder)
  - User Tower
    - 입력 데이터 (Input Features)
      - User Query: "비 오는 날 강아지랑 캠핑 갈건데 필요한 거 추천해줘"
      - LLM Processing: 입력 쿼리에서 핵심 키워드/의도 추출
        - 결과: "Rainy, Camping, Dog, Outdoor"
    - 모델 구조 (Architecture)
      - Text Encoder: LLM을 사용하여 숫자로 벡터화 ⇒ User Vector
  - Item Tower
    - 입력 데이터
      - 1. Text Features
        - a. *user review*: title (제품명), text (**LLM Processed Reviews**, "0|제품이 유용한 상황/키워드 추출")
        - b. *item metadata*: features, description, details (제품 설명)  
→ [Title] + [Review Keywords] + [Features] + [Description] + [Details] + [Description Average Pooling Vector] + [Title Average Pooling Vector]
      - 2. Categorical, Numerical Features (Optional)
        - a. categories, store, average\_rating, price
    - 모델 구조 (Architecture)
      - Text Encoder: 합쳐진 텍스트를 LLM에 통과시켜 숫자로 벡터화 ⇒ Item Vector
  - 작동 방식
    - Faiss를 사용하여 User Vector와 가장 유사한(Cosine Similarity) Item Vector 100개 추출 (Item Vector를 미리 계산하여 Faiss에 저장)
- LLM
  - Llama3.1:8B : query to keyword
  - BAAI/bge-m3 : keyword

## Step 2: Re-ranking (최종 순위)

- Model (XGBoost/LightGBM)
- Features
  1. cosine similarity
  2. item metadata (*price, average\_rating, review\_count* 등)
  3. 리뷰 반응: *helpful\_votes*

## Step 3: Explanation (LLM Generation) → API

최종 선정된 5개의 아이템에 대해, 사용자가 입력한 시나리오에 맞춰 “왜 이 제품이 당신에게 딱인지” 설명하는 텍스트를 생성

- 프롬프트 전략 (RAG): **(1) 유저 시나리오, (2) 아이템 메타데이터, (3) “핵심 리뷰”**를 사용하여 근거 제시

### ▼ 프롬프트 예시

당신은 전문 쇼핑 어드바이저입니다.

사용자의 [시나리오]에 맞춰 추천 제품을 소개하는 매력적인 설명을 작성하세요.

#### [입력 정보]

1. 사용자 시나리오: "{user\_scenario}"
2. 제품명: "{item\_title}"
3. 제품 특징: "{item\_features}"
4. 실제 구매자 리뷰 요약: "{review\_summary}" (또는 도움이 된 리뷰 원문)

#### [작성 가이드]

- 사용자의 시나리오(상황)와 제품의 특징을 연결해서 설명하세요.
- 단순히 스펙을 나열하지 말고, 실제 리뷰 내용을 인용하여 신뢰도를 높이세요.
- 3문장 이내로 간결하고 설득력 있게 작성하세요.

#### [출력 예시]

"비 오는 날 캠핑을 계획 중이시군요! 이 텐트는 3000mm 방수 코팅이 되어 있어 폭우에도 끄떡없습니다. 실제 리뷰에서도 '밤새 비가 왔지만 내부가 뾰송했다'는 평이 많아 안심하고 사용할 수 있습니다."

## Model Evaluation

- 데이터 나누기 (**Data Splitting**) - “시간” 기준, *sort\_timestamp*로 **Time-based Split**
  - 전체 데이터: 시간순 정렬
  - Train Set (과거 80%)**: 모델 학습용
  - Validation Set (중간 10%)**: 학습 중 성능 체크 & 하이퍼파라미터 튜닝용
  - Test Set (최근 10%)**: 최종 성능 평가용

**평가 방식 시뮬레이션:** Test Set에 있는 리뷰 하나를 가져온다. “이 리뷰 텍스트(시나리오)를 입력했을 때, 사용자가 실제로 구매했던 그 아이템(asin)을 맞췄는가?”를 테스트

- 1단계 평가: Retrieval → **Recall@100**
- 2단계 평가: Re-ranking → **NDCG@5**

## 4. 기술 스택 (Tech Stack)

**Language:** Python

**Deep Learning:** PyTorch, Hugging Face Transformers (BERT) ?

**Vector Search:** Faiss

**DB:** ChromaDB ? MySQL ?

**LLM:** HyperCLOVA X, LangChain

**Data Processing:** Pandas, NumPy

김종민

