

**SK네트웍스 Family AI 과정 14기  
 데이터 전처리 학습된 인공지능 모델**



| **산출물 단계** | 데이터 전처리 |
| --- | --- |
| **평가 산출물** | 학습된 인공지능 모델 |
| **제출 일자** | 2025-08-29 |
| **깃허브 경로** | https://github.com/skn-ai14-250409/SKN14-Final-2Team |
| **작성 팀원** | 박빛나 |

1. **모델 목적**: 사용자의 자연어 향 설명에서 적절한 향수 메인 어코드를 분류하기 위한 모델
2. **모델 아키텍처 설계**

* 선정 모델: xlm-roberta-base기반 분류 모델
* 아키텍처 개요:

| **계층** | **구성 요소** | **역할** |
| --- | --- | --- |
| 입력층 | XLM-RoBERTa 토크나이저(SentencePiece) | 한/영(다국어) 문장을 서브워드 단위 토큰으로 변환 |
| 임베딩 | 토큰 임베딩 + 위치 임베 | 토큰 ID를 연속 벡터로 투영, 순서 정보 부여 |
| 인코더 | Transformer Encoder × 12층 (다중 헤드 자기어텐션 + FFN, 잔차/정규화) | 문맥적 의미 표현 학습, 장문/다의어 문맥 해소 |
| 풀링 | [CLS] 토큰 표현 (또는 mean pooling) | 문장 수준 표현 추출 |
| 분류 헤드 | Dropout → Linear (num\_labels) | 라벨 점수 로짓 산출 |
| 출력 | Sigmoid (라벨별) | 각 라벨의 존재 확률(0~1) 산출 → 다중라벨 결정(Threshold) |

* 아키텍처 시각화: (구조 다이어그램 삽입 또는 링크)
* 설계 근거
  + xlm-roberta-base 선택 이유
  + 강력한 다국어 일반화 성능
  + 형 잡힌 파라미터 규모(성능↔자원 트레이드오프)
  + 문맥 이해력과 장문 처리 능력
  + 사전학습 표현의 전이 학습 효율
  + 다중라벨 분류와의 궁합
  + 생태계/툴체인 지원

1. **모델 학습 요약**

* 학습 데이터 수: 26319건 → 빈 라벨 264행 제거 : 26055건
* 검증 데이터 수: 23449건
* 평가 데이터 수: 2606건
* 성능 평가 결과:
  + 성능 평가 지표(F1, Jaccard) 사용 이유 : 다중 라벨 분류 모델로 단순 정확도의 한계
    - Micro-F1 : 전체 라벨/샘플을 통틀어 TP/FP/FN을 한 번에 합산해서 계산
      * 자주 등장하는 ‘강라벨(헤드)’의 영향이 큼 → 전반적인 운영 지표로 자주 씀
    - Macro-F1 : 라벨별 F1의 산술 평균(모든 라벨을 동일 가중)
      * 드문 ‘꼬리 라벨(테일)’에 취약하면 크게 떨어짐. 값이 낮다는 건 라벨 불균형 영향이 큼을 시사
    - Jaccard : 샘플 단위로 (예측 ∩ 정답) / (예측 ∪ 정답)을 계산 후 평균
      * 예측 라벨 집합이 정답 집합과 전체적으로 얼마나 일치하는지 평가

| **Epoch** | **Training Loss** | **Validation Loss** | **Micro F1** | **Macro F1** | **Jaccard** |
| --- | --- | --- | --- | --- | --- |
| 1 | 0.206600 | 0.202550 | 0.467879 | 0.154158 | 0.363232 |
| 2 | 0.187400 | 0.196308 | 0.496100 | 0.171883 | 0.404976 |
| 3 | 0.183900 | 0.196595 | 0.524571 | 0.186756 | 0.428780 |
| 4 | 0.180400 | 0.200093 | 0.515051 | 0.213191 | 0.423484 |
| 5 | 0.156100 | 0.202451 | 0.533556 | 0.218313 | 0.444474 |

* 일반화 성능 평가 (검증기반) :  
  + 전체 데이터셋을 학습/검증(90/10)으로 분할하여 검증 기반 일반화 성능을 평가
  + 학습 중 매 에폭마다 검증셋에 대해 Micro F1, Macro F1, Jaccard(samples) 지표를 산출하고, 해당 지표 중 Micro F1을 기준으로 최적 체크포인트를 선택
  + 검증셋의 예측 확률을 활용해 라벨별 임계값(Threshold) 최적화를 수행함으로써 라벨 간 불균형 및 다중 정답 특성을 반영하도록
  + 학습·검증 곡선에서 큰 괴리 없이 수렴하여 과소적합 징후는 없고 과적합은 통제된 것으로 판단
  + 일반화 성능 확인을 위해서는 독립적인 Test set 평가를 추가로 수행할 예정

1. **저장 및 배포**

* 저장 형식:

| **항목** | **설명** |
| --- | --- |
| 저장 파일명 | xlmr\_model.pt |
| 저장 형식 | PyTorch .pt 파일 |
| 저장 방법 | 가장 마지막 체크 포인트 경로에서 state\_dict() 만 저장 → .pt 파일  import torch  from transformers import AutoModelForSequenceClassification  # 가장 마지막 체크포인트 경로  CKPT\_DIR = r"C:\Users\Playdata2\Desktop\hf\_xlmr\_ckpt\checkpoint-14660"  OUTPUT\_PATH = r"C:\Users\Playdata2\Desktop\xlmr\_model.pt"  # 모델 불러오기  model = AutoModelForSequenceClassification.from\_pretrained(CKPT\_DIR)  # state\_dict()만 저장 → .pt 파일  torch.save(model.state\_dict(), OUTPUT\_PATH) |

* 모델 사양 요구 사항:  
  + 프레임워크: **PyTorch 2.x** (Hugging Face Transformers·Datasets 기반)
  + **GPU/CPU 호환**:
* GPU(권장): 학습 시간 및 대용량 배치 처리에 유리 (CUDA 11+)
* CPU: 학습은 느리지만 **추론은 가능**  
  + 환경 설정 :
    - torch==2.1.0
    - transformers==4.38.0
    - datasets==2.18.0
    - accelerate==0.29.3
    - scikit-learn==1.4.2
    - pandas==2.2.2
    - numpy==1.26.4
    - tqdm==4.66.4
* 모델 테스트:  
  + 모델 적재 및 추론 테스트 진행중
  + Inference 예시:  
      
     입력: "깨끗하게 빨래하고 말린 상쾌한 향"  
     출력: ['Citrus', 'Classical', 'Fresher']

1. **종합 평가 및 활용 방안**

* 모델 안정성
  + 고정 시드 및 동일 전처리 파이프라인에 기반하여, 저장/로드 반복 시 성능 변동이 매우 작음
* 일반화 가능성
  + 검증셋 기준 Micro F1 / Macro F1 / Jaccard를 매 epoch 산출하고, load\_best\_model\_at\_end=True 설정으로 최적 체크포인트를 선택
  + 검증 예측 확률을 활용한 라벨별 임계값 최적화(0.30~0.70 탐색)를 수행하여 다중라벨·불균형 특성을 반영
* 재사용성 / 배포 용이성
  + 모델 아티팩트는 Hugging Face 표준 포맷(save\_pretrained)으로 저장되어 배포·로딩이 간단
* 향후 활용 계획
  + API 서버에 탑재
  + 챗봇 응답 분류
  + 유사 문장/향 검색
  + RAG 파이프라인 보조

1. **추가 기재**

* 저장된 모델 파일 위치 또는 URL: https://drive.google.com/file/d/1TZXiwiX-xcg01b5J3K6DhNpQkawhRiEy/view