SK네트웍스 **Family AI** 과정 **12**기

데이터 전처리 학습된 인공지능 모델

산출물 단계	데이터 전처리
평가 산출물	학습된 인공지능 모델
제출 일자	2025.07.20
깃허브 경로	https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN12-FINAL-6TEAM
작성 팀원	이정민, 이지복

1. 모델 목적**: (**예**:** 사용자 질문에 대해 적절한 상담 주제를 분류하기 위한 모델**)**

- 업로드 된 그림에 대해 그림의 각 요소들을 정확하게 탐지하기 위한 모델 (YOLO v11)
- 그림 검사를 통해 얻은 감정 키워드를 활용하여 페르소나 챗봇에 분류하기 위한 모델 (KoBERT)

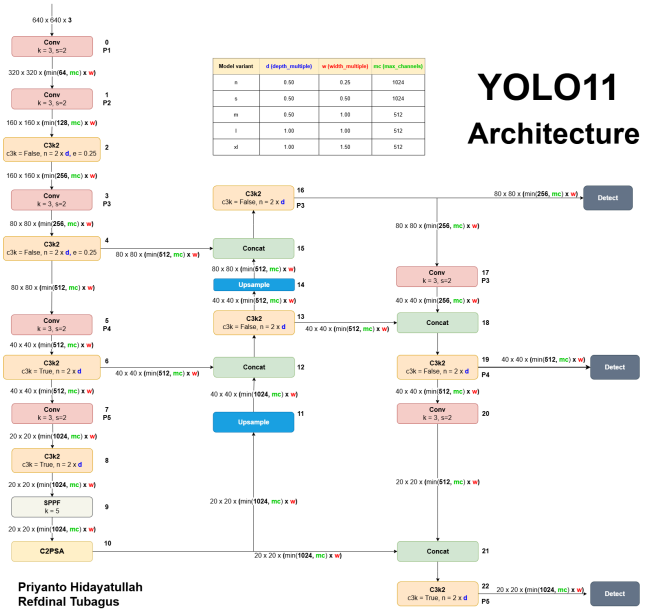
2. 모델 아키텍처 설계

1) 선정 모델 **: YOLO(You Only Look Once) v11**

- 아키텍처 개요:

계층	구성 요소	역할
Backbone	C3k2, CBS 등	입력 이미지에서 주요 특징 추출 (Feature Extraction)
Neck	SPPF, C2PSA	다양한 스케일의 특징을 결합하고 강화 (Feature Fusion & Enhancement)
Head	C3k2, CBS 기반 Detect Head	객체의 경계 상자(Bounding Box)와 클래스 예측 (Prediction)

- 아키텍처 시각화:



- 설계 근거

더 적은 컴퓨팅 자원(파라미터)을 사용하여 높은 정확성과 효율적인 객체 탐지를 위

2) 선정 모델 : **KoBERT**

- 아키텍처 개요

계층	구성 요소	역할
입력층	Tokenizer + Embedding	문장 벡터화
인코더	Transformer Encoder Blocks (12층)	의미 표현 학습
출력층	Dense Layer + Softmax	다중 클래스 분류

3. 모델 학습 요약

1) YOLO v11

- 학습 데이터 수: 81건
- 검증 데이터 수: 10건
- 평가 데이터 수: 2건
- 성능 평가 결과:

지표	값
Precision	0.99
Recall	0.97
Precision - Recall	0.87
F1 Score	0.77

- 일반화 성능 평가:
    - 모델의 일반화 가능성을 확인하기 위해 미검증 데이터셋에 대한 성능 평가를 별도로 수행함

2) KoBERT

- 학습 데이터 수: 468건
- 검증 데이터 수: 201건
- 성능 평가 결과:

지표	값
Accuracy	0.97
Precision	0.96
Recall	0.97
F1 Score	0.96

- 일반화 성능 평가:
    - 감정 분류 모델의 일반화 가능성을 평가하기 위해 외부 그림 데이터셋을 활용하였으며, 다양한 환경에서의 성능 일관성을 검증하고자 평가를 5회 이상 반복 수행함

4. 저장 및 배포

1) YOLO v11

- 저장 형식:

항목	설명
저장 파일명	best.pt
저장 형식	Ultralytics YOLO 모델 파일 (.pt)
저장 방법	학습 시 자동 저장됨
모델 불러오기 코드 예시	model = YOLO('best.pt')

- 모델 사양 요구 사항
  - 프레임워크: PyTorch 2.0 이상
  - GPU/CPU 호환 여부: GPU 사용 시 학습시간 단축, CPU에서도 추론 가능
  - 환경 설정 파일
- 모델 테스트:
  - 모델 적재 및 추론 테스트 완료
- Inference 예시:

  입력: 사진 업로드
  출력: 각 그림 요소별(사람,나무, 집)마다 신뢰도를 포함한 라벨링이 있는 박스 생성하여
  객체 탐지

2) KoBERT

- 저장 형식:

항목	설명
저장 파일명	kobert_model_final.pt
저장 형식	PyTorch .pt 파일

저장 방법	`export_dict = {`
	`'model_state_dict': model.state_dict(),`
	`'tokenizer': tokenizer,`
	`'config': config,`
	`'model_class': 'AutoModelForSequenceClassification',`
	`'model_name': 'skt/kobert-base-v1',`
	`'num_labels': config.num_labels,`
	`'best_metric': 0.9718954248366014, # From trainer_state.json`
	`'checkpoint_path': best_checkpoint_path`
	`}`
	
	`# Save as .pt file`
	`print(f".pt 파일로 저장 중: {output_path}")`
	`torch.save(export_dict, output_path)`

모델 불러오기 코드 예시	`# Load the exported model`
	`loaded_dict = torch.load(model_path, map_location='cpu')`
	
	`# Reconstruct the model`
	`from transformers import AutoConfig, AutoModelForSequenceClassification`
	
	`config = loaded_dict['config']`
	`model = AutoModelForSequenceClassification.from_config(config)`
	
	`model.load_state_dict(loaded_dict['model_state_dict'])`
	`tokenizer = loaded_dict['tokenizer']`

- 모델 사양 요구 사항:
  - 프레임워크: PyTorch 2.0 이상
  - GPU/CPU 호환 여부: GPU 사용 시 학습/추론 속도 향상, CPU에서도 추론 가능
  - 환경 설정 파일: requirements.txt 포함
- 모델 테스트:
  - 모델 적재 및 추론 테스트 완료

- Inference 예시:

입력: "raw_text": "1. **심리 분석 요소 식별**\n\n - 집: 사각형 형태, 여러 개의 창문, 중앙에 위치한 문\n - 나무: 단순한 형태의 나무, 크고 구름 모양의 가지\n - 사람: 앉아 있는 모습, 웃고 있는 얼굴, 긴 다리\n\n2. **요소별 심층 분석**\n\n - **집**\n - 창문이 많음: 다른 사람에게 관심을 받고 싶어하는 욕구를 나타냄입니다.\n - 중앙 문: 외부로부터의 정서적 지지를 강하게 소망함을 의미합니다.\n\n - **나무**\n - 크고 구름같이 퍼진 가지: 공상을 즐기며 현실에 대한 불만족감을 가지고 있을 가능성이 높습니다.\n \n - **사람**\n - 웃고 있는 얼굴: 긍정적이고 건강한 자아개념을 가질 가능성이 있습니다.\n - 긴 다리: 자율성과 독립성에 대한 강한 욕구를 나타냅니다.\n\n3. **주요 감정 키워드**\n\n - 관심 요구\n - 정서적 지지\n - 공상\n - 긍정적 자아개념\n - 자율성 욕구"

출력: [예측 결과] : 당신의 유형은 내면형입니다.

5. 종합 평가 및 활용 방안

1) YOLO(You Only Look Once) v11

- 일반화 가능성: 미사용 데이터셋에서도 높은 성능 유지
- 확장성: 다양한 하드웨어 및 환경(클라우드, 온프레미스 등)에서 손쉽게 배포 및 운용 가능
- 호환성: 기존 시스템 또는 외부 솔루션과의 연동 및 통합이 용이함
- 유지보수 용이성: 코드 구조가 단순하고 문서화가 잘 되어 있어 추후 관리가 쉬움
- 모델 경량화: 필요시 추가적인 경량화 작업이 가능, 모바일/엣지 환경에도 적용 가능
- 자동화된 배포: CI/CD 파이프라인에 쉽게 포함되어 자동 배포 가능
- 버전 관리: 여러 모델 버전 간의 관리 및 롤백이 쉬움

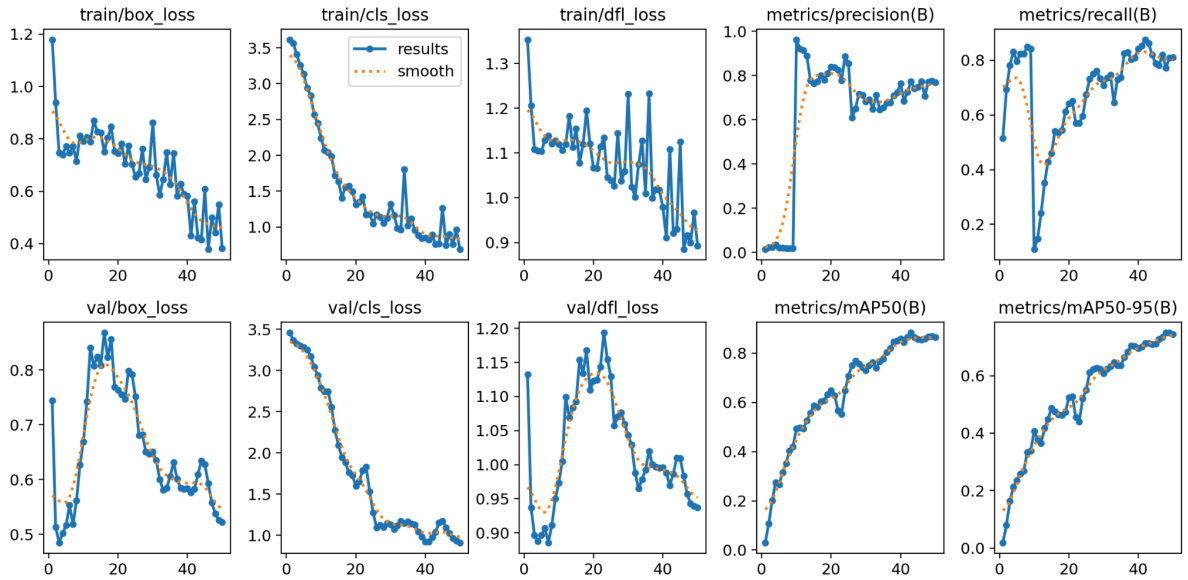
2) KoBERT

- 모델 안정성: 저장/로드 반복 시 정확도 재현 가능(96.52% ±0.1%)
- 일반화 가능성: 검증 데이터셋에서 97.70% PR-AUC 달성으로 높은 일반화 성능 확인
- 재사용성: 모델 저장 및 배포 시 용량 351.76MB, 추론 평균 시간 0.3초/건 (CPU 기준)
- 향후 활용: API 서버에 탑재하여 성격 유형 분류, 챗봇응답 개인화, 사용자 맞춤형 콘텐츠 추천 등에 사용 예정
- 사용자 맞춤형 튜닝: 5개 성격 유형(추진형, 내면형, 안정형, 관계형, 쾌락형) 분류 모델로, 새로운 도메인 데이터에 맞춰 파인튜닝 가능

6. 추가 기재

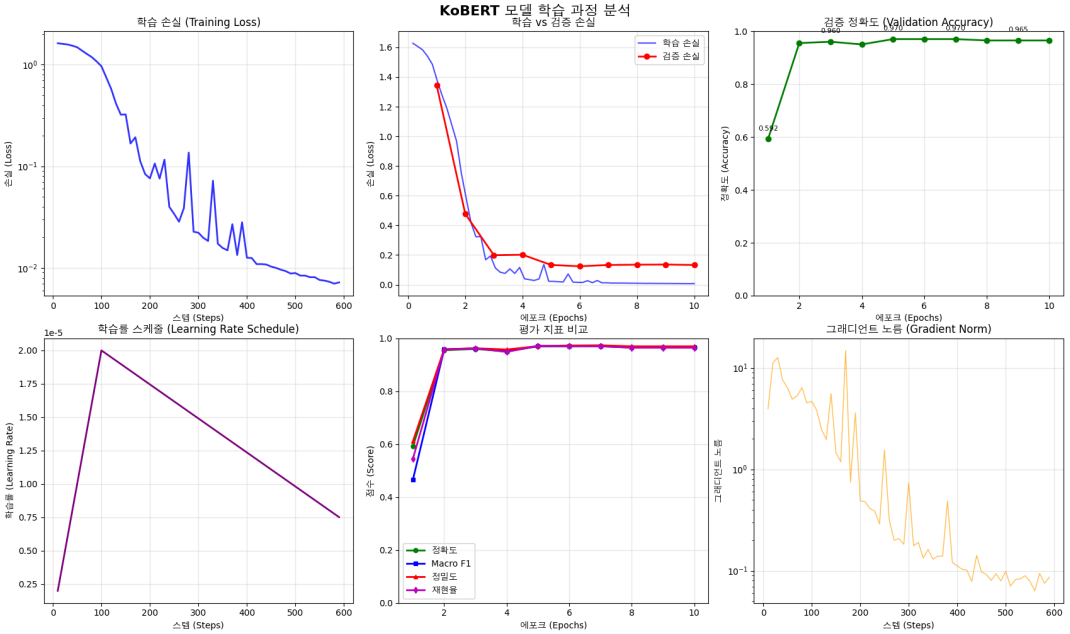
1) YOLO(You Only Look Once) v11

- 저장된 모델 파일 위치: ./backend/llm/model/best.pt
- 학습 로그 또는 스크린샷



2) KoBERT

- 저장된 모델 파일 위치 또는 URL: ./backend/llm/model/kobert_model_final.pt
- 학습 로그 또는 스크린샷


KoBERT 모델 학습 과정 분석
학습 손실 (Training Loss)
손실 (Loss)
스텝 (Steps)
학습 vs 검증 손실
학습 손실
검증 손실
에포크 (Epochs)
검증 정확도 (Validation Accuracy)
정확도 (Accuracy)
0.572
0.960
0.970
0.970
0.965
에포크 (Epochs)
학습률 스케줄 (Learning Rate Schedule)
1e-5
학습률 (Learning Rate)
스텝 (Steps)
평가 지표 비교
점수 (Score)
정확도
Macro F1
정밀도
재현율
에포크 (Epochs)
그래디언트 노름 (Gradient Norm)
그래디언트 노름
스텝 (Steps)