

**SK네트웍스 Family AI 과정 12기  
 데이터 전처리 학습된 인공지능 모델**



| **산출물 단계** | 데이터 전처리 |
| --- | --- |
| **평가 산출물** | 학습된 인공지능 모델 |
| **제출 일자** | 25.07.18. |
| **깃허브 경로** | <https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN12-FINAL-3TEAM> |
| **작성 팀원** | 박슬기 |

1. **모델 목적**

회의의 발화들을 분류하여 회의록 작성에 필요한 발화(유효 발화)와 필요 없는 발화(무효 발화)를 걸러내는 것을 목적으로 하는 이진 분류 모델

1. **모델 아키텍처 설계**

* 선정 모델: Klue / BERT-base 모델
* 아키텍처 개요:

| **계층명** | **역할** | **구성 요소** |
| --- | --- | --- |
| Embedding | 입력 문장을 벡터화 | WordPiece Tokenizer  (AutoTokenizer.from\_pretrained) + 사전학습된 임베딩 (BERT 모델) |
| Encoder | 의미 표현 학습 | Transformer Encoder (Multi-Head Attention + FFN + Dropout(0.3)) |
| Classification Head | 태그 예측 | Linear Layer (Dense Layer) + Softmax  (출력층 자동 포함됨: AutoModelForSequenceClassification) |

* 아키텍처 시각화:



* 설계 근거  
  본 모델은 회의 발화가 회의록 제작에 필요한 내용인지 유/무효를 이진 분류하는 것을 목표로 함.

[TGT],[/TGT] 토큰으로 감싸여진 target 문장의 유/무효를 문맥에 맞게 판단하게 하기 위해 target 문장의 앞 뒤 발화를 하나의 문장으로 합쳐 해당 내용이 회의록 작성에 필요한 내용인지 문맥에서 확인하도록 함.

임베딩된 발화를 Linear layer를 통해 logits를 계산하고, softmax 함수를 이용하여 0 또는 1 값을 반환하도록 강제함.

1. **모델 학습 요약**

* 학습 데이터 수: 8,000건
* 검증 데이터 수: 2,000건
* 평가 데이터 수: 2,400건
* 성능 평가 결과:

| **지표** | **값** |
| --- | --- |
| Accuracy | 87% |
| Precision | 91.5% |
| Recall | 89.1% |
| F1 Score | 90.3% |

* 일반화 성능 평가:  
  + Train, Val, Test 데이터셋에 중복되는 발화문이 없도록 제작하여 미검증 데이터셋(Test set)에 대한 성능 평가 결과 포함
  + 과적합 방지를 위해 조기 종료, Dropout 등 사용
  + 과소적합 발생 없음 (훈련 정확도와 테스트 정확도 유사)

1. **저장 및 배포**

* 저장 형식:

| **항목** | **설명** |
| --- | --- |
| 저장 파일명 | Ttalkkak\_model\_v1.pt |
| 저장 형식 | PyTorch .pt 파일 |
| 저장 방법 | torch.save(model.state\_dict(), 'Ttalkkak\_model\_v1.pt') |
| 모델 불러오기 코드 예시 | model.load\_state\_dict(torch.load('Ttalkkak\_model\_v1.pt')) |
| 저장 위치 | [Ttalkkak\_model\_v1](https://drive.google.com/drive/folders/1Mg46BJyXC46ua0UV30C7VfS3aAR_hR1p?usp=sharing) |

* 모델 사양 요구 사항:  
  + 프레임워크: PyTorch 2.0
  + **GPU** 사용 시: 학습 속도 대폭 단축 가능
  + **CPU** 환경에서도: 모델 로드 및 추론(인퍼런스) 지원
  + 환경 설정 파일: requirements.txt 포함
* 모델 테스트:  
  + 모델 적재 및 추론 테스트 완료
  + Inference 예시:  
      
     입력: “이 내용은 누구 담당이죠?”  
     출력: “1” (예측 정확)

1. **종합 평가 및 활용 방안**

* 모델 안정성: 저장/로드 반복 시 정확도 재현 가능 (±0.1%)
* 일반화 가능성: 미사용 데이터셋에서도 높은 성능 유지
* 향후 활용: Ai 모델과 함께 Runpod에서 실행 예정. 회의록 문장 1차 분류에 사용

1. **추가 기재**

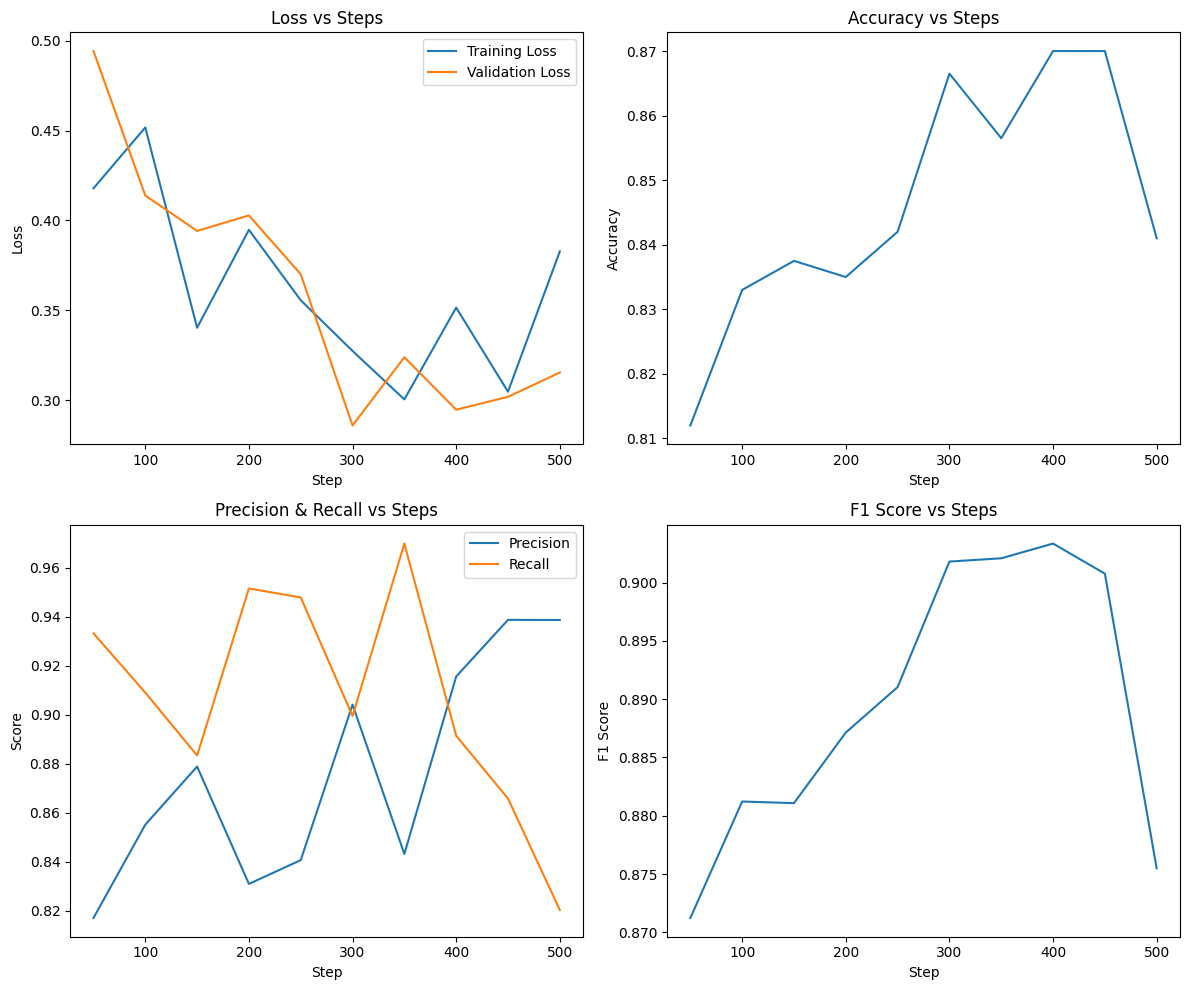
* 저장된 모델 파일 위치 또는 URL:[Ttalkkak\_model\_v1](https://drive.google.com/drive/folders/1Mg46BJyXC46ua0UV30C7VfS3aAR_hR1p?usp=drive_link)
* 모델 저장/불러오기 코드:

model = AutoModelForSequenceClassification.from\_config(config)

state = torch.load("Ttalkkak\_model\_v1.pt", map\_location="cpu")

model.load\_state\_dict(state)

model.eval()

* 학습 로그 또는 스크린샷
* Train log
* Classification-report

