

**SK네트웍스 Family AI 과정 12기  
 데이터 전처리 인공지능 학습 결과서**



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

**1. 모델 비교 및 선정 이유**

* 비교 대상 모델:

| **모델명** | **종류** | **선정 이유** |
| --- | --- | --- |
| beomi/KcBERT | Transformer 기반  사전학습 모델 | * 네이버 댓글을 학습에 사용, 일반적인 구어체에 최적화 된 모델이라 실제 대화 구분에 유효할 것이라 판단하여 선정[[1]](#footnote-0) |
| klue/bert | 한국어 특화 BERT | * 국립 국어원, 나무위키, 뉴스 크롤링 등 다방면의 한국어 데이터를 사용한 대표적 범용 한국어 bert 모델[[2]](#footnote-1) * 공식문서, 뉴스, 회의록 등 정형화된 한글 데이터 처리에 탁월하여 선정함.[[3]](#footnote-2) |
| monologg/  koelectra-small-v3 | Koelectra 소형 모델 | * 한국어로 학습된 ELECTRA 소형 모델로서 제한된 컴퓨팅 환경에서의 성능 비교를 위해 선정[[4]](#footnote-3) * "Transformers 라이브러리 호환성을 통해 실제 배포 환경에서 활용 가능한 모델로서 실용적 비교 기준 제공¹ * ELECTRA 기반의 효율적인 pre-training 방식을 통해 작은 모델 크기 대비 우수한 성능을 보이는 것으로 알려져 있어, 리소스 효율성 측면에서의 벤치마킹에 적합[[5]](#footnote-4) |
| monologg/  koelectra-base-v3-discriminator | Koelectra 기본 모델 | * KoELECTRA 계열 중 상대적 대용량 모델(110M 파라미터)로서 Small 모델(14M 파라미터) 대비 성능 차이 분석을 위해 선정[[6]](#footnote-5) * Discriminator 구조를 활용하여 다양한 한국어 자연어 처리 태스크에서의 성능 비교에 적합[[7]](#footnote-6) * 동일한 ELECTRA 아키텍처 내에서 모델 크기에 따른 성능 차이를 분석하기 위한 대조군 역할 * 한국어 도메인에서 충분한 검증을 거친 모델로서 신뢰성 있는 비교 기준 제공 |

* 실험 모델 수: 총 4종
* 최종 선정 모델: Klue/BERT

**2. 모델 구조 및 아키텍처**

2.1 모델 아키텍처 도식

* 입력층 → 임베딩층 → 인코더블록(N개) → 분류기 출력층 (Softmax)

2.2 구성 요소 설명:

| **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) |

특이사항

* Tokenizer 특수 토큰 추가 : [TGT], [/TGT] 추가 후 모델 리사이즈 (tokenizer.add\_special\_tokens), 문장 구조 정보를 강조
* 출력층 num\_label = 2로 설정 : 소프트맥스 기반 이진 분류
* 최적화 : AdamW 옵티마이저 + ReduceLROnPlateau 스케줄러

**3. 학습 설정 및 하이퍼파라미터**

| 학습 데이터 수 | Train 8000건 + Val 2000건 | |
| --- | --- | --- |
| 검증 데이터 수 | Test 2000건 + Sample 400건 | |

| **항목** | **Beomi & Klue / Bert 모델** | **KoELECTRA 모델** | | |
| --- | --- | --- | --- | --- |
| **small** | | **base** |
| **기준** | **최적** |
| 에폭(Epoch) 수 | 5 | 5 | 7 | 5 |
| 배치 크기 (Batch Size) | 16 | 16 | 32 | 16 |
| 학습률 (Learning Rate) | 5e-5 | 5e-5 | 1e-4 | 5e-5 |
| 옵티마이저 | AdamW (torch.optim.AdamW) | AdamW | AdamW | AdamW |
| 러닝레이트 스케줄러 | ReduceLROnPlateau  (mode=’max’, factor=0.5, patience=1) | ReduceLROnPlateau | | ReduceLROnPlateau  (factor=0.5, patience=2) | | --- | | ReduceLROnPlateau |
| 손실 함수 | CrossEntropy  Loss | CrossEntropy  Loss | CrossEntropy  Loss | CrossEntropy  Loss |
| 조기 종료 기준 | 검증 F1 개선 없을 시 2회 | 검증 F1 개선 없을 시 2회 | 검증 F1 개선 없을 시 3회 | 검증 F1 개선 없을 시 2회 |
| 평가 전략 | Steps 단위 평가  (매 50 스텝, eval\_steps=50) | Steps 단위 평가  (매 50 스텝, eval\_steps=50) | Steps 단위 평가  (eval\_steps=100) | Steps 단위 평가  (매 50 스텝, eval\_steps=50) |
| 모델 저장 전략 | Steps 단위 저장  (매 50 스텝, save\_steps=50)  최대 2개 체크포인트 보관 (save\_total\_limit=2) | Steps 단위 저장  (매 50 스텝, save\_steps=50)  최대 2개 체크포인트 보관 (save\_total\_limit=2) | Steps 단위 저장  (save\_steps=100)  최대 2개 체크포인트 (save\_total\_limit=2) | Steps 단위 저장  (매 50 스텝, save\_steps=50)  최대 2개 체크포인트 보관 (save\_total\_limit=2) |
| 로그 기록 주기 | 10 스텝마다  (logging\_steps=10) | 10 스텝마다  (logging\_steps=10) | 20 스텝마다 (logging\_steps=20) | 10 스텝마다  (logging\_steps=10) |

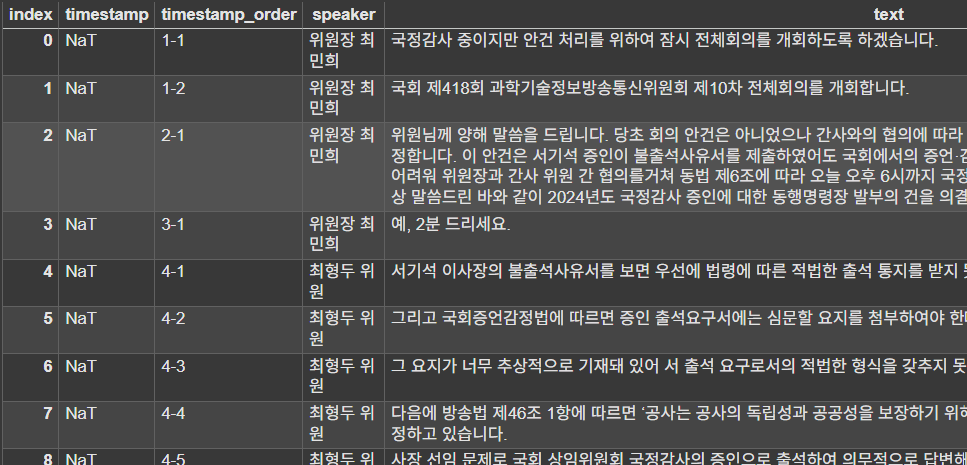
* monologg/koelectra-small-v3-discriminator는 작은 모델이어서 표현력이 제한적이므로 다른 학습 전략으로 학습 진행

3.1 학습 데이터 준비 과정

1. 국회 데이터 및 영문 데이터 번역본 전처리

발화자의 발화문이 긴 경우 각 문장별로 추론하기 위하여 발화문을 문장부호를 기준으로 구분함.

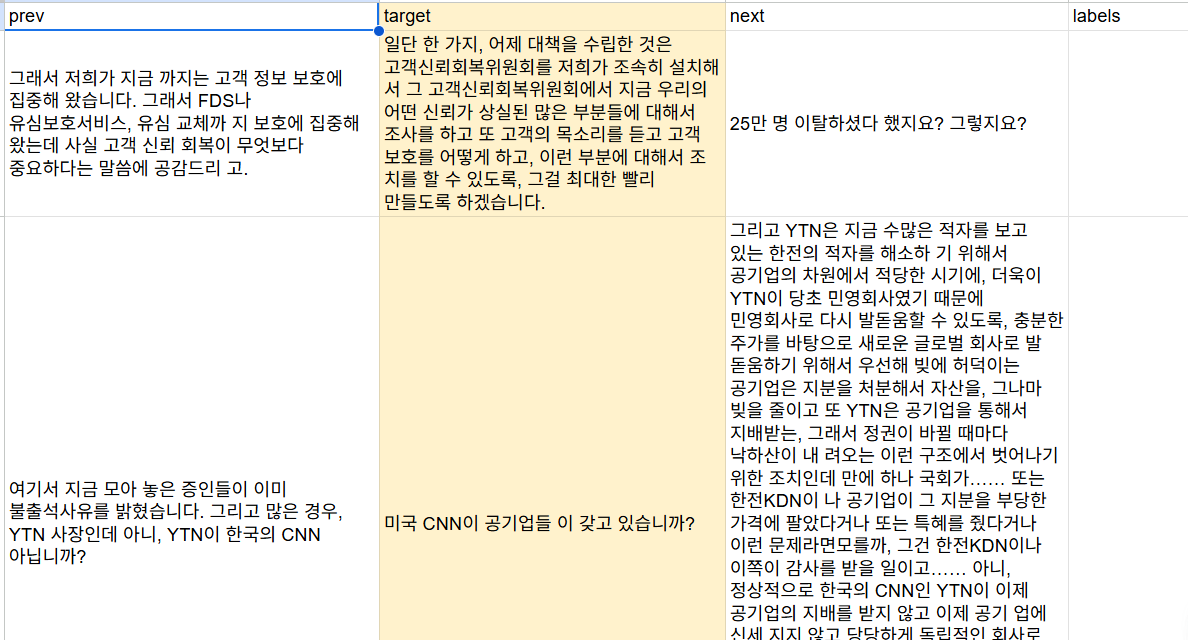
발화문이 나눠진 경우 speaker를 확인하기 어려우므로 타임스탬프 컬럼의 순서를 기준으로 타임 스탬프 오더 컬럼을 추가하여 speaker를 찾을 수 있도록 오더링함.



1. 타겟 데이터 준비

전체 라인들 중 랜덤한 문장을 target 문장으로 선택함.

문맥을 확인할 수 있도록 Target 문장의 앞의 2 문장을 prev 문장, 뒤의 2 문장을 next 문장으로 포함, 라벨링을 위한 labels 컬럼 추가.



확보 데이터 개수 : 국회 회의록 데이터 200개 + 영어 번역 데이터 200개

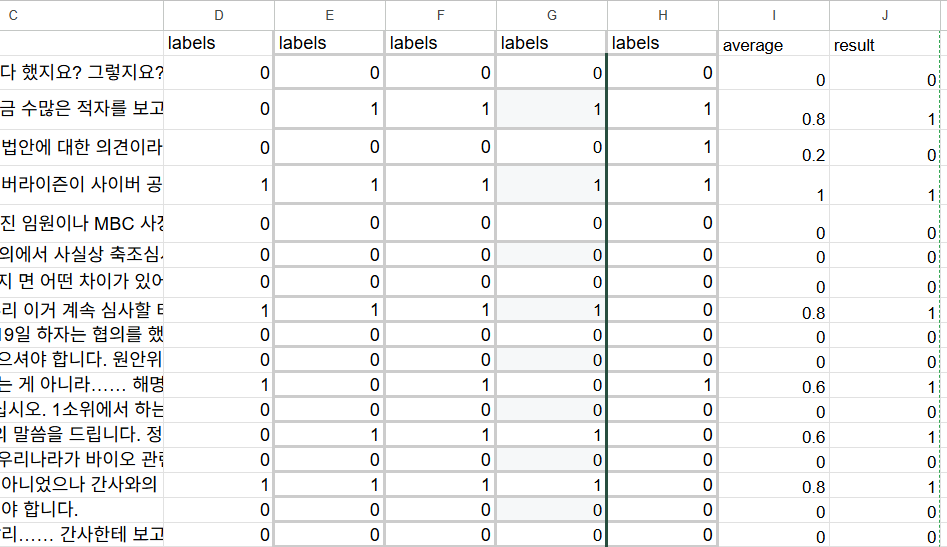
→ 자연스러운 한글 회의 표현 + IT 관련 언어 습득을 위해 반반씩 확보

1. 수동 라벨링
2. 라벨링 기준 정의

라벨링 기준 : [라벨링 기준](https://docs.google.com/spreadsheets/d/1suPHQMNamo8hPQcFA4LI36CbKp-Y5mkJcMwLdPynQhc/edit?gid=0#gid=0)

랜덤으로 확보한 400개의 데이터를 라벨링 기준에 따라 팀원 각자 수기로 라벨링함.

1. 수동 라벨링 데이터 취합 및 정답 라벨링



팀원 각자의 라벨링 값의 평균값을 기준으로 라벨링값을 0과 1로 구분

다수결의 원리에 따라 더 많은 사람이 선택한 방향을 라벨링 값의 정답으로 처리함

1. GPT-4.1 프롬프트 및 퓨샷 엔지니어링
2. 프롬프트/퓨샷 엔지니어링

취합한 400개의 데이터의 라벨링을 지운 후 GPT-4.1 모델에 추론 진행

총 5회에 걸쳐 프롬프트 및 퓨샷 엔지니어링 시도

프롬프트를 변경하고 실제 예시를 들어 설명하는 등 다양한 방향으로 시도함

1. 퓨삿 프롬프트 선정 과정

1차 결과 (프롬프트 엔지니어링) : [GPT-4.1추론결과ver01](https://docs.google.com/spreadsheets/d/1RP-wsaxlvSsfE3goAtBv8jQbBffuuFo9Wu5bJz9P1R8/edit?usp=sharing)

| 전략 | - 유효 발화 선정 기준을 프롬프트로 넣어 판단 |
| --- | --- |
| GPT 정답 일치율 | 0.75 |

2차 결과 (프롬프트 엔지니어링) : [GPT-4.1추론결과ver02](https://docs.google.com/spreadsheets/d/1Op5hammGmZj-a2XHgQepnsYnqFJmbbZcFHdjoZYAXEs/edit?usp=sharing)

| 전략 | - 유효 발화 선정 기준에 예시를 프롬프트에 추가 |
| --- | --- |
| GPT 정답 일치율 | 0.78 |

3차 결과 (퓨샷) : [GPT-4.1추론결과ver03](https://docs.google.com/spreadsheets/d/1_M_E4TKWNwk8nrCIAMakLmUuHqk-x3nbdEpg5hc25Ak/edit?usp=sharing)

| 전략 | - 실제 데이터의 예시를 퓨샷으로 적용 |
| --- | --- |
| GPT 정답 일치율 | 0.87 |

4차 결과 (퓨샷) : [GPT-4.1추론결과ver04](https://docs.google.com/spreadsheets/d/1Q4VLLp3T3O35Y4IDSxf8XOSe77-xMO0eFxh6Sos5PKE/edit?usp=sharing)

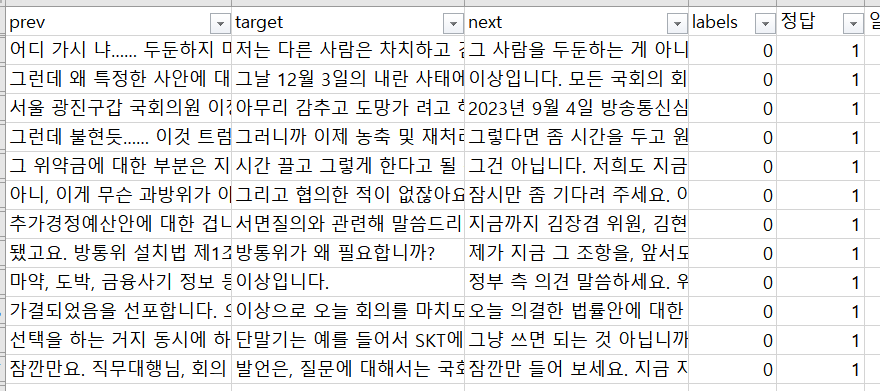
| 전략 | - GPT가 1~3차에 걸쳐 항상 다른 답변을 냈던 예시들 중 확실히 유효/무효한 표현을 구체적 예시로 제시함.  - 결정 후 재확인 프롬프트 추가 |
| --- | --- |
| GPT 정답 일치율 | 0.94 |

5차 결과 (퓨샷) : [GPT-4.1추론결과ver05.xlsx](https://docs.google.com/spreadsheets/d/1q2XlxLbEqWRoT5PWHyJwqU1WrknDhFwA/edit?usp=sharing&ouid=116634207080490419599&rtpof=true&sd=true)

| 전략 | - 본 분류가 회의록 작성을 위한 것임을 명시  - 4차에서 틀린 예시들 중 확실한 유효/무효 예시를 Reason 내용을 추가하여 이유를 학습하도록 함. |
| --- | --- |
| GPT 정답 일치율 | 0.97 |



7회차 퓨샷 엔지니어링에서 수기 라벨링과 GPT-4.1라벨링의 일치율이 최고값 0.97에 도달 (400개 데이터 중 불일치 데이터 12개)



불일치 데이터 12개 개별 확인

* 해당 문장들이 회의록 작성을 기준으로 유효한 데이터일 수 있다고 판단
* 유효한 문장이지만 무효로 처리한 경우가 발생하지 않았으므로 해당 프롬프트를 신뢰하기로 판단함.

1. 라벨링 데이터 확보

해당 프롬프트를 기준으로 총 12,000개의 데이터 라벨링 작업 진행

GPT-4.1의 레이트 리미트로 인해 라벨링 작업 지연 발생

지연을 최소화하고 라벨링 중복 작업을 피하기 위해 12,000개의 데이터를 1000개씩 12개 파일로 청크하여 라벨링 진행

1. Train, val, test 데이터 구분

라벨링한 총 12,400개의 데이터셋 구분

| 이름 | 데이터 수 | 용도 | 비고 |
| --- | --- | --- | --- |
| Train\_data | 8,000 | 모델 학습용 데이터 |  |
| Val\_data | 2,000 | 학습 확인용 데이터 |  |
| Test\_data | 2,000 + 400 | 모델 학습 검증용 데이터 | 수기 샘플 데이터 포함 |

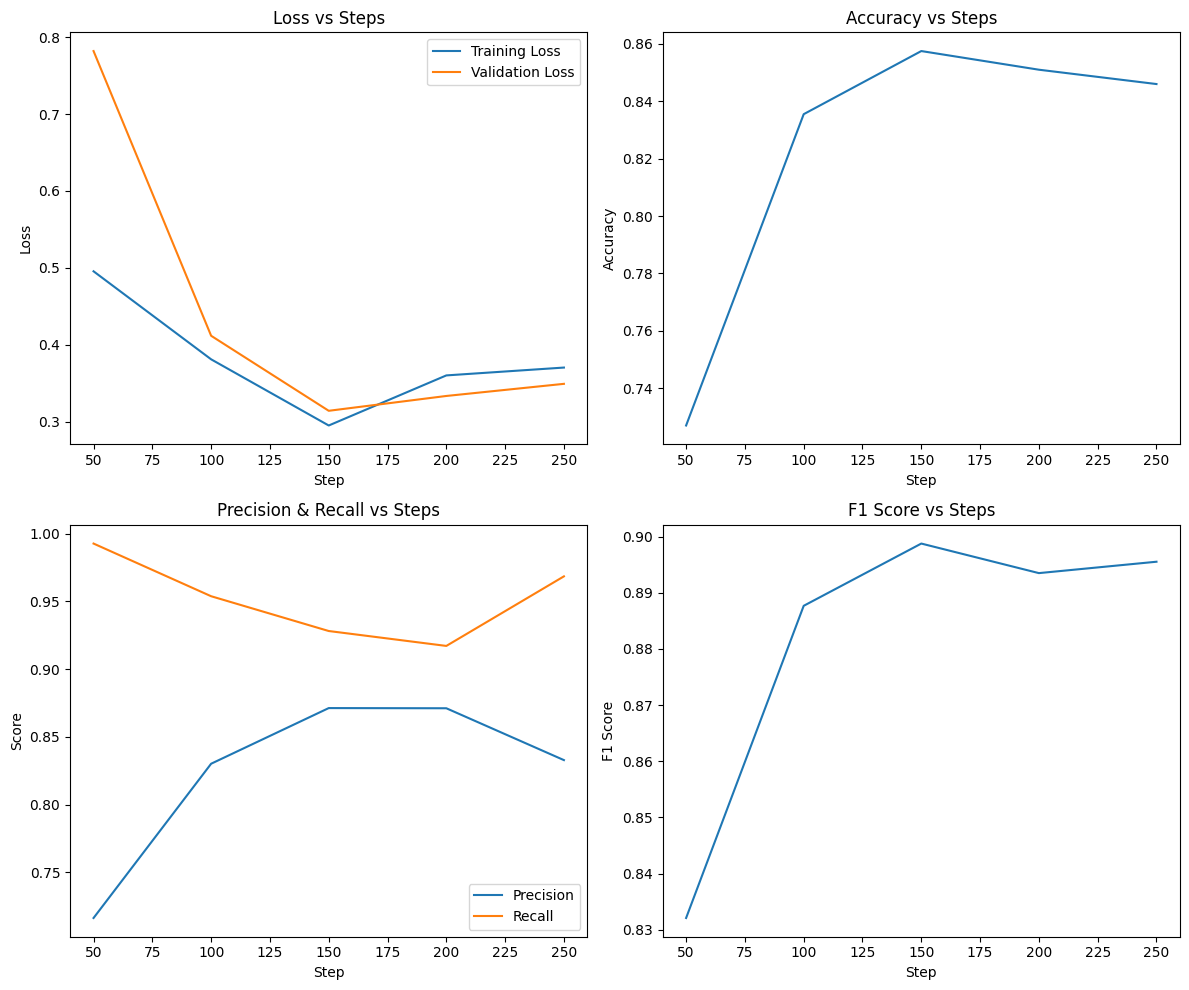
**4. 학습 결과 및 성능 평가**

4.1. 학습 결과 요약

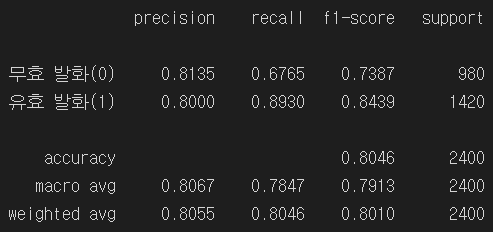
| **모델** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| --- | --- | --- | --- | --- |
| beomi/KcBERT | 0.8575 | 0.8712 | 0.9281 | 0.8987 |
| klue/bert | 0.8700 | 0.9155 | 0.8914 | 0.9033 |
| Koelectra small  (기본 파라메터) | 0.8200 | 0.8336 | 0.9192 | 0.8743 |
| Koelectra small  (최적 파라메터) | 0.8350 | 0.8530 | 0.9156 | 0.8832 |
| Koelectra base | 0.8275 | 0.8472 | 0.9112 | 0.8780 |

4.2. 그래프 : 학습/검증 loss 변화, accuracy 변화 등

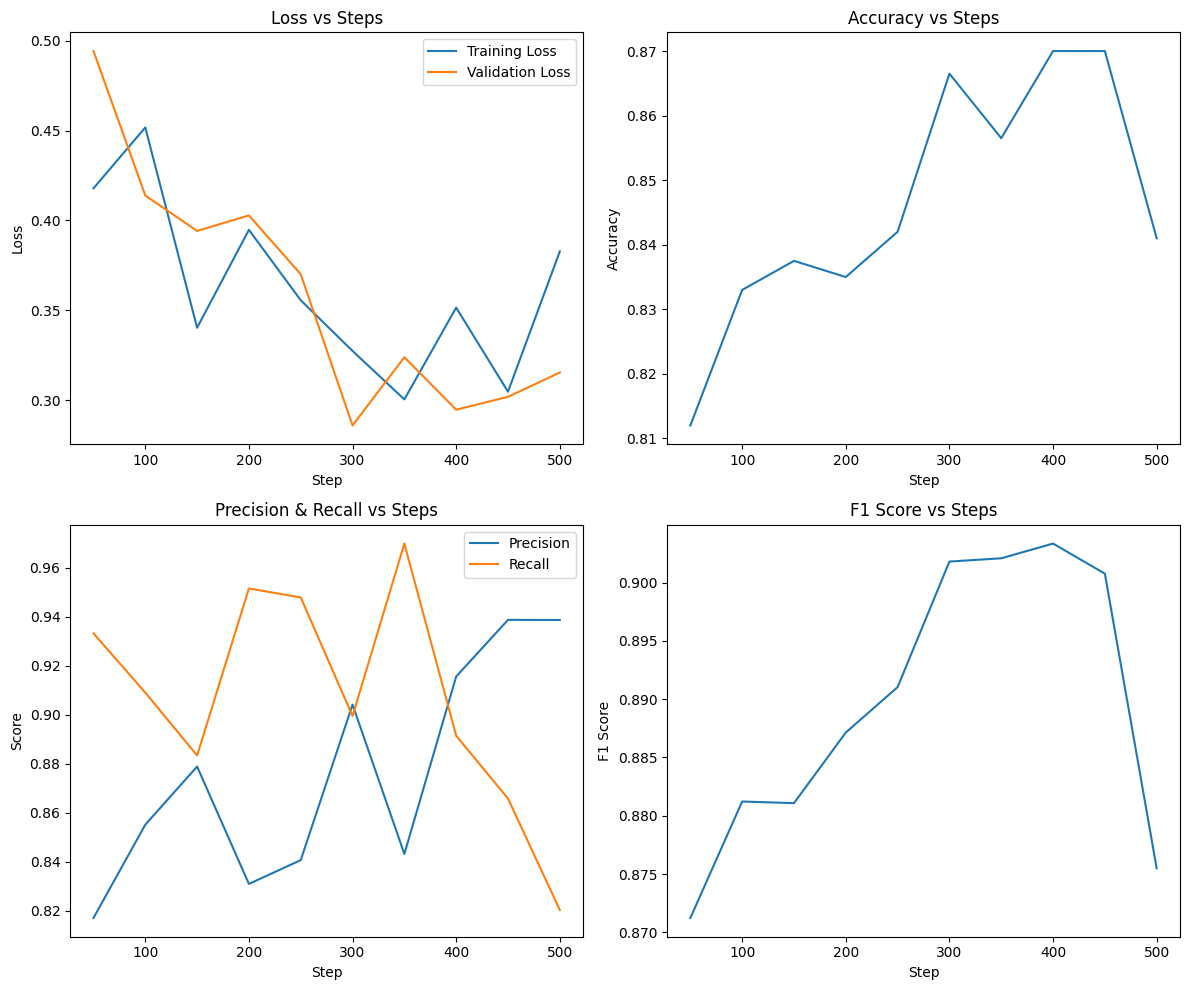
4.2.1 beomi/KcBERT : 150 step에서 최고점수, 0.5 epoch에서 조기 종료



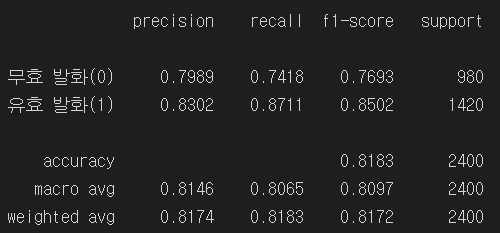
* classification-report



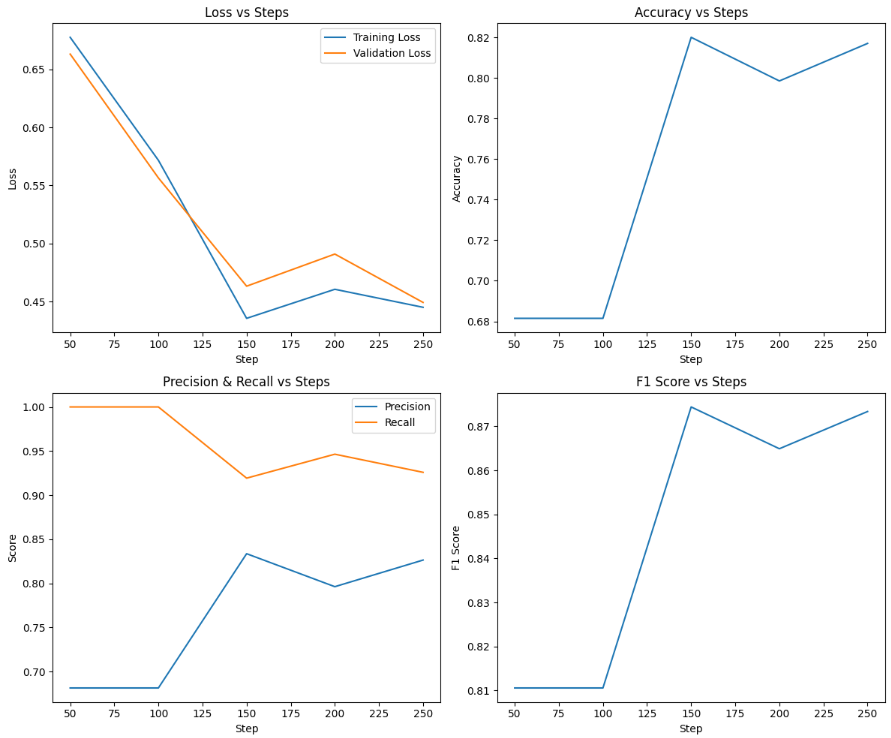
4.2.2 klue/bert : 400 steps에서 최고점수, 1 epoch에서 조기종료



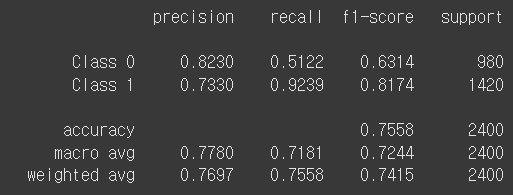
* classification-report



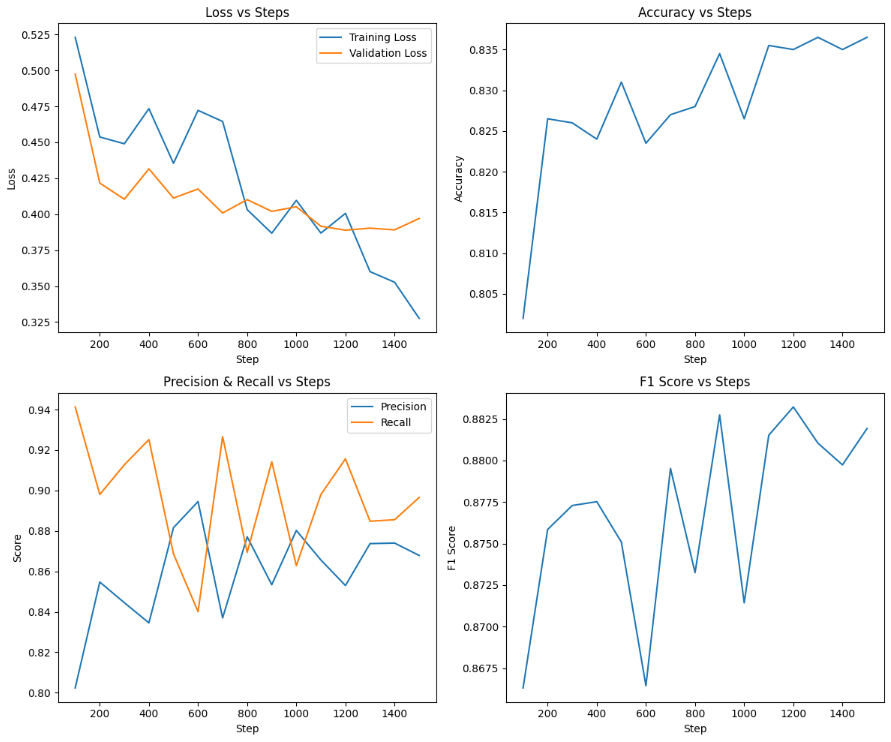
4.2.3 - 1 KoELECTRA-small(기준 파라미터) : 150 steps에서 최고점수, 0.5 epoch에서 조기종료



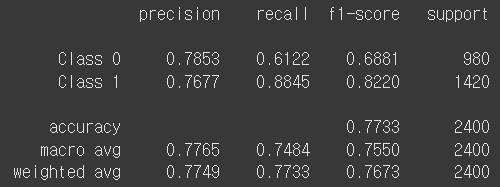
* Classification-report

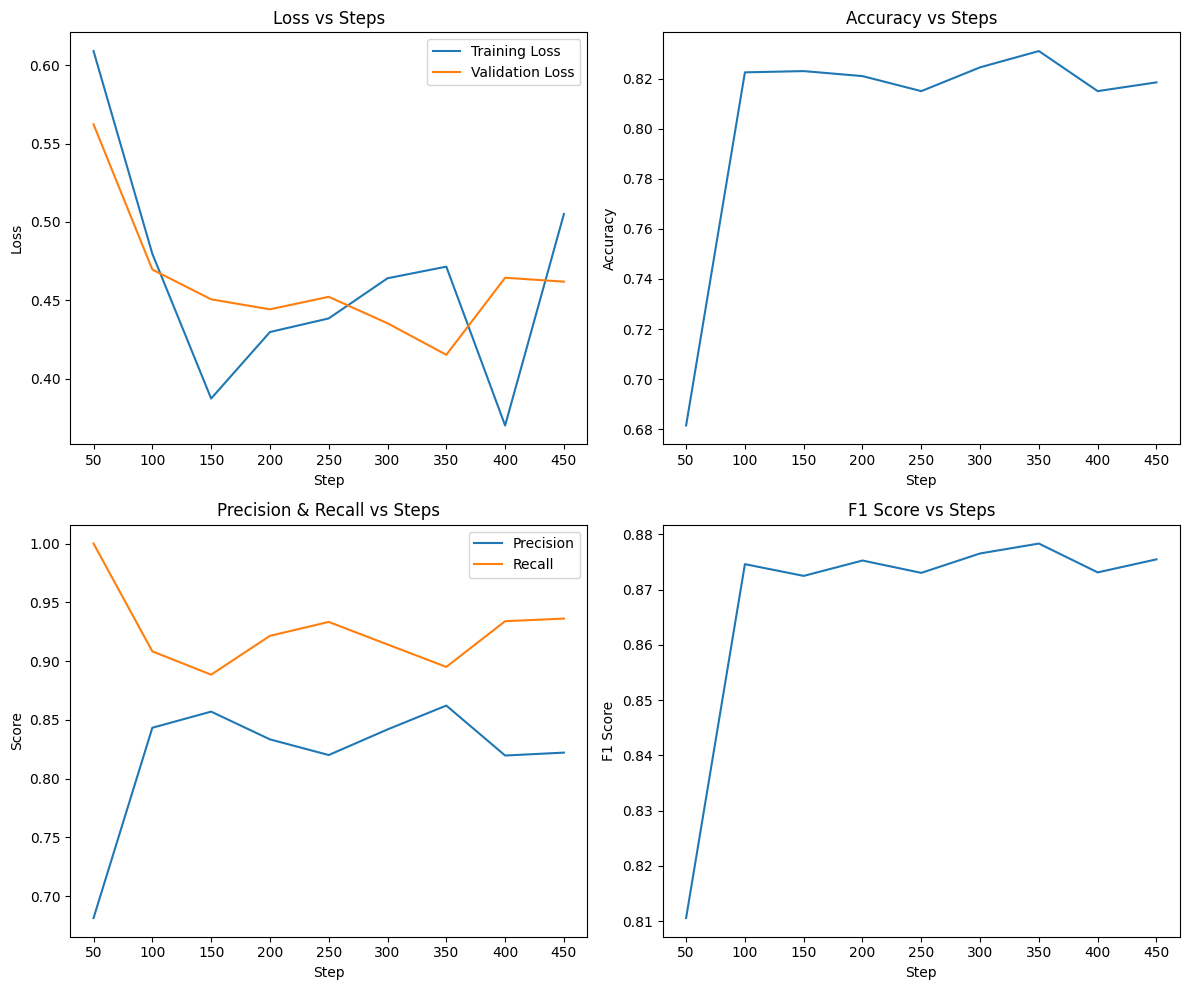


4.2.3 - 2 KoELECTRA-small(최적 파라미터) : 1200 steps에서 최고점수, 6 epoch에서 조기종료

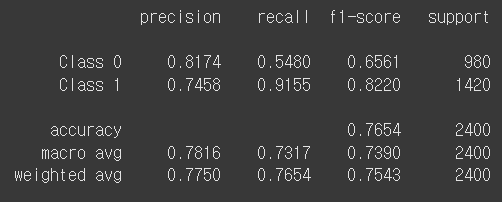


* Classification-report



4.2.4 KoELECTRA-base : 300 steps에서 최고점수, 0.8 epoch에서 조기종료

* Classification-report



4.3 해석 및 분석

* 데이터 구조
  + 모델 입력 및 출력 구조
    - 타겟 문장에 [TGT] [/TGT] 스페셜 토큰 추가
    - 타겟문장 앞 2문장, 뒤 2문장을 timestamp-order를 기준으로 추가
      * 비어있는 경우 예외처리를 통하여 “ “로 반환
    - 모델의 타겟 문장의 유효성 평가
    - 모델의 라벨링을 기준으로 라벨 0인 데이터만 남겨 반환함
    - 라벨 1인 데이터의 로그 기록 남김
      * 디버깅 및 재학습용
* 유효 발화를 가려내는 recall 수치는 beomi/kcbert 모델이 더 높게 나왔지만, **전반적인 성능은 klue/bert 모델이 0.5~4% 나은 수치**를 보임
* 다만 beomi/kcbert의 경우 무효 발화 recall 수치가 너무 낮은 관계로 **실제 적용에서 무효발화를 찾아내는데에 어려움이 발생할 것으로 예상**됨
* 발화의 유무효를 판단하고 회의록 작성에 필요없는 무효 발화를 걸러내는 모델 목적의 특성상 **밸런스있는 모델을 선택하는 것이 유리할 것으로 판단**하여 **최종적으로 Klue/bert-base 모델을 선택**

**5. 과적합/과소적합 대응**

* 적용 기법:

| **기법** | **설명** | **적용 여부** |
| --- | --- | --- |
| Dropout | 과적합 방지 | O  (hidden\_dropout\_prob=0.3, attention\_probs\_dropout\_prob=0.3) |
| 조기 종료 (Early Stopping) | 성능 저하 시 종료 | O (EarlyStoppingCallback(patience=2) 사용) |
| 학습률 감소 | Plateau 시 자동 감소 | O (ReduceLROnPlateau(mode='max', factor=0.5, patience=1) 사용) |
| 교차 검증 | 데이터 분산 고려 | X  (중복되지 않는 데이터셋으로 train, val set 구분, 단일 홀드아웃 방식으로 학습 및 검증 진행행) |

**6. 결론 및 향후 계획**

* 최종 선정 모델: Klue/BERT
* 활용 방안: 데이터 문장 분류, 필요 토큰 절약
* 향후 계획:  
  + 실제 데이터로 실사용 환경 평가 예정
  + 실제 데이터를 이용한 결과에 따라 테스트 데이터를 추가하여 재훈련 고려
  + 삭제된 문장은 디버그 로그에 남겨 수동 확인 필요

**7. 부록**

* 학습 코드 경로 또는 GitHub 링크
  + [klue-bert 테스트(wandb 포함).ipynb](https://colab.research.google.com/drive/1bdMA-O1FuKMpzgM1bmIaCNRp5AFUv4XW?usp=sharing)
* 주요 파라미터 설정 config.json 또는 코드 스니펫
  + [klue-bert-best](https://drive.google.com/drive/folders/1rtOKWD-tQa2IhSfF8NNieT57ryp3smmi?usp=sharing)
* 사용 데이터셋
  + [dataset](https://drive.google.com/drive/folders/1982_ksElAtQ73prbZfcXtVG_YokjbLud?usp=sharing)

**8. 변경 이력**

| 변경일 | 변경자 | 변경 내용 | 비고 |
| --- | --- | --- | --- |
| 2025-07-15 | 박슬기 | - | 초안 작성 |
| 2025-07-18 | 박슬기 | 내용 전면 수정 |  |

1. Hugging face beomi/kcbert-base 공식 페이지 <https://huggingface.co/beomi/kcbert-base> [↑](#footnote-ref-0)
2. KLUE: Korean Language Understanding Evaluation <https://arxiv.org/abs/2105.09680> [↑](#footnote-ref-1)
3. Hugging face klue/bert-base 공식 페이지 <https://huggingface.co/klue/bert-base> [↑](#footnote-ref-2)
4. Park, J. (2020). KoELECTRA: Pretrained ELECTRA Model for Korean. GitHub repository. - <https://github.com/monologg/KoELECTRA> [↑](#footnote-ref-3)
5. <https://medium.com/dair-ai/bert-is-extremely-inefficient-this-is-how-to-solve-it-688b09350f10> - "ELECTRA introduces a pre-training framework that enables BERT-small's GLUE performance to be achieved with the same size model for 12x less compute" [↑](#footnote-ref-4)
6. Korean PLM Models Comparison. GitHub repository. https://github.com/sooftware/Korean-PLM [↑](#footnote-ref-5)
7. Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In International Conference on Learning Representations (ICLR). [↑](#footnote-ref-6)