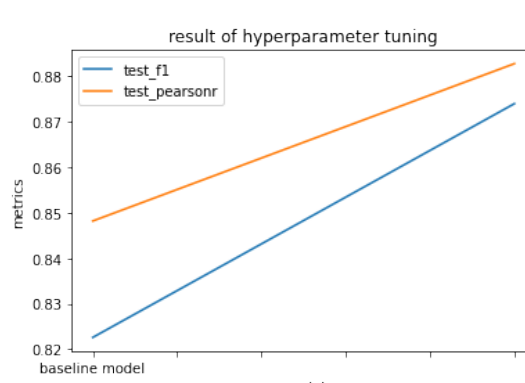


기업과제 3 학습과정 보고서

7팀 조하진

	보고 내용												
제목	<ul style="list-style-type: none">NLU 문장 유사도 계산 (STS)												
담당한 역할	<ul style="list-style-type: none">모델 선택과 모델링, 하이퍼파라미터 튜닝, 만든 모델로 문장유사도 방법 search												
학습 방법 설명	<div>1) 모델 선택<ul style="list-style-type: none">Hugging face의 사전학습된 klue/roberta-base 모델을 사용해 fine tuningbert에 더 큰 배치사이즈로 더 많은 데이터를 학습시킨 모델, Dynamic masking을 수행하여 문장 유사성 부분에서 SOTA를 도달하였기에 한국어 데이터로 사전훈련된 roberta 선택</div> <div>2) 파라미터 튜닝<ul style="list-style-type: none">optuna 패키지를 이용하여 hyper parameter searchepoch : earlystopping으로 score 변화 없을시 종료되게 수행random seed : grid search로 탐색해 좋은 값을 내주는 seed를 얻음학습률은 스케줄러를 사용하는 것과 고정값을 주는 2가지 방식으로 실험배치사이즈, weight_decay 탐색 수행</div> <div>3) 훈련 과정<div>I. 데이터 전처리<ul style="list-style-type: none">train 데이터 train, valid로 splitroberta tokenizer를 이용하여 토큰화 및 모델 입력에 맞게 변환 작업</div><div>II. finetuing<ul style="list-style-type: none">metric으로 f1 score과 pearson 사용baseline : 파라미터를 조정하지 않고 default 값으로 돌린 모델로 설정, 기본적인 모델의 메트릭을 산출해서 어느방향으로 나아가야할지 기본으로 삼기 위함</div><div>III. 하이퍼 파라미터 튜닝<ul style="list-style-type: none">상위 기재한 내용으로 여러가지 실험을 거쳐 세개의 모델로 후보군을 좁히고 F1과 피어슨이 골고루 높은 첫번째 모델로 최종 모델 선택(하단의 table1, 2 참조)</div><div>IV. Evaluate & Predict<ul style="list-style-type: none">test set으로 평가 및 예측 진행</div><div>V. 최종모델로 restAPI 제작 (하단의 이미지 참조)</div></div> <div>4) 결과<table><thead><tr><th></th><th>f1</th><th>pearson</th><th>loss</th></tr></thead><tbody><tr><td>baseline</td><td>0.82</td><td>0.85</td><td>0.49</td></tr><tr><td>최종모델</td><td>0.87</td><td>0.88</td><td>0.35</td></tr></tbody></table></div>		f1	pearson	loss	baseline	0.82	0.85	0.49	최종모델	0.87	0.88	0.35
		f1	pearson	loss									
baseline	0.82	0.85	0.49										
최종모델	0.87	0.88	0.35										

결과해석
및
보완할점

- 하이퍼파라미터 튜닝
 - 적은 배치사이즈를 사용했을 경우 테스트셋과 스코어가 차이가 덜 났던것으로 보아 적은 배치 사이즈가 모델의 일반화에 도움이 된다는 것을 알 수 있었다
 - 스케줄러를 사용한 것이 학습률을 고정으로 주는 것보다 결과가 더 좋았다
 - Weigh decay를 주는것과 안주는 것에도 점수차이가 꽤 있었다
 - 랜덤시드에 따라서도 값이 많이 달라져서 시드 고정의 중요성에 대해 알게 되었다
- 긍정/부정의 평가 부재
 - 단어가 비슷하면 긍정/부정의 뜻이 달라도 비슷한 문장이라고 인식
 - 단순히 문장 자체를 토큰화해서 비교하는 방식이어서 발생하는 문제로 보인다
- 데이터 셋을 더 늘리거나 노이즈를 추가해서 학습하여 모델에 다양성 주기
- 유사도 뿐만 아니라 다양한 metric을 이용해서(rouge 등..) semantic적인 유사도도 파악 할 수 있는 모델로의 발전

- Optuna1
 - learning rate: 1.4654261946051012e-05
 - batch_size: 8
 - weight_decay: 0.061673419363888114
 - warmup_steps: 0.01
- A 모델:
 - lenarning rate: iner schedule
 - batch_size = 32,
 - weight decay: 0.01
- B 모델
 - learning rate: 5e-5
 - batch_size: 64
 - weight_decay: 0.01

table 1 : 세가지 후보 모델

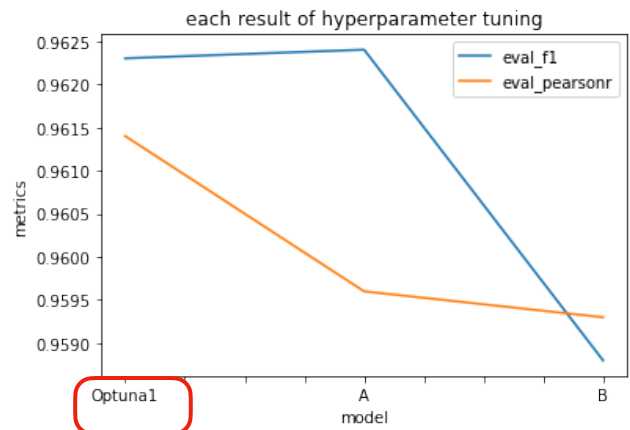


table 2 : optuna 선택

