[기업과제 3] - 개인 보고서

9팀 강민지

1. 담당한 역할

1) [리서치] 한국어 STS task에 사용될 수 있는 모델, baseline code 서치
2) [모델링] 기본 baseline code로 KoELECTRA 모델의 성능 확인
3) [모델링] Inference 코드 작성
4) [파라미터 튜닝] Hyperparameter tuning 및 성능 결과 저장 코드 작성
5) [파라미터 튜닝] batch_size=16에서 hyperparameter tuning 진행 및 결과 공유

2. 모델 선정 배경 및 이유

먼저, KLUE 팀에서 발표한 논문 Park et al.(2021)[1]에서 KLUE-STS 데이터셋을 사용하여 다양한 모델의 성능을 이미 검증한 바 있으므로, 해당 논문 내용을 바탕으로 STS task에 높은 성능을 보였던 모델을 조사하였다. 이때, 코랩(Google Colab)의 computing power로 LARGE 모델은 돌아가지 않는 문제가 있었기 때문에 LARGE 모델을 제외하고 성능 기준 상위 4개 모델을 선정하였다. 나아가, KLUE 논문에서는 SKTBrain에서 개발한 KoBERT 모델을 사용하지 않았기 때문에, KoBERT도 포함하여 아래의 5개 모델을 최종 후보군으로 선정하였다.

- 모델 후보: ① KLUE-RoBERTa-base, ② KLUE-RoBERTa-small, ③ KoELECTRA-base
  ④ KLUE-BERT-base, ⑤ KoBERT

Hyperparameter를 epochs=3, batch_size=16, learning_rate=3e-5 로 설정한 뒤, baseline code에서 위의 5개 후보 모델에 대한 성능을 확인하였다. 교차 검증(k_fold=5)을 사용하여 과적합되지 않은 모델의 평균 성능을 측정한 결과는 아래와 같다.

모델 이름	평균 성능(Pearson's R)
KLUE-RoBERTa-base	0.9665
KLUE-RoBERTa-small	0.9646
KoELECTRA-base	0.9682
KLUE-BERT-base	0.9540
monologg/KoBERT	**0.9861**

이 중에서 monologg/KoBERT 모델의 성능이 가장 좋은 것으로 측정되었기에, KoBERT 모델로 최종 선정하고 hyperparameter tuning을 진행하였다.

[1] Park, S., Moon, J., Kim, S., Cho, W. I., Han, J., Park, J., ... & Cho, K. (2021). KLUE: Korean language understanding evaluation. arXiv preprint arXiv:2105.09680.

3. Hyperparameter Tuning 과정 및 결과

KoBERT는 BERT 모델을 한국어 데이터셋으로 학습시킨 모델이다. BERT 논문(Devlin et al.(2018)[2]) 에서 hyperparameter에 대한 range를 아래와 같이 제안하고 있기 때문에, 본 프로젝트에서 사용하는 STS 모델의 hyperparameter range 또한 동일하게 설정하고 tuning을 진행하였다.

Hyperparameter	Range
Number of Epochs	2, 3, 4
Learning Rate	2e-5, 3e-5, 5e-5
Batch Size	16, 32

Hyperparameter에 따른 평균 성능은 아래의 표와 같다.

batch_size(lr) \ epochs	2	3	4
16(2e-5)	Pearson: 0.9938 F1: 0.9811	Pearson: 0.9966 F1: 0.9851	Pearson: 0.9980 F1: 0.9870
16(3e-5)	Pearson: 0.9947 F1: 0.9780	Pearson: 0.9973 F1: 0.9837	Pearson: 0.9980 F1: 0.9870
16(5e-5)	Pearson: 0.9946 F1: 0.9800	Pearson: 0.9968 F1: 0.9807	Pearson: 0.9979 F1: 0.9879
32(2e-5)	Pearson: 0.9494 F1: 0.9372	Pearson: 0.9921 F1: 0.9777	Pearson: 0.9968 F1: 0.9830
32(3e-5)	Pearson: 0.9816 F1: 0.9721	Pearson: 0.9951 F1: 0.9835	**Pearson: 0.9978** **F1: 0.9881**
32(3e-5)	Pearson: 0.9884 F1: 0.9748	Pearson: 0.9961 F1: 0.9881	Pearson: 0.9976 F1: 0.9846

가장 좋은 성능을 보인 모델은 epochs=4, batch_size=32, learning_rate=3e-5 로 학습시킨 모델이었다. 따라서 최종 모델로는 해당 환경에서 마지막 교차검증 폴드, 마지막 에폭(즉, 4번째 에폭)에서 학습시킨 체크포인트 모델을 사용하였다.

4. 훈련 과정

[모델 정보]

- Pretrained Model: monologg/kobert
- Loss Function: MSELoss
- Optimizer: AdamW
- Learning Rate: 3e-5
- Epochs: 4
- Evaluation Metric: F1 Score, Pearson's R

[훈련 과정]

1) KLUE-STS train dataset에 대해 불용어 제거, 특수기호 제거를 통해 기본적인 전처리를 거친다.
2) Tokenizer는 KoBertTokenizer, pre-trained model은 huggingface에서 제공하는 'monologg/kobert'

[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

모델을 사용하였고, pre-trained model 위에 하나의 layer만 더 쌓아 최종적으로 문장 간 유사도를 logit 값으로 return하는 CustomRegressor 모델을 생성하였다.

3) k_fold를 5로 설정한 교차 검증을 통해 매 fold마다 train dataset 내에서 train, valid set을 분리하고 이를 통해 학습 및 평가를 반복하여 5개 fold의 평균 성능을 측정한다.
4) Hyperparameter tuning을 거쳐 평균 성능을 비교하고, 가장 좋은 성능을 낸 모델을 최종 체크포인트 모델로 사용한다.

5. 최종 결과 분석

최종 체크포인트 모델로 dev dataset을 평가한 결과는 다음과 같다.

Metric	Score
Accuracy	0.8478
Recall	0.7809
Precision	0.8909
F1 Score	0.8323
Pearson's R	0.8839

Base-sized model 기준으로 KLUE 팀이 STS task에 대해 발표한 최고 성능은 F1 Score 85.73, Pearson's R 92.5 정도였지만 본 프로젝트에서 생성한 모델은 그에 미치지는 못하는 성능을 보였다. 현재 CustomRegressor에서 한 개의 layer만 추가하였기 때문에, 신경망의 구조를 더 복잡하게 하였을 때 성능이 개선되는지 볼 필요가 있겠다는 생각이 들었다.