**기업과제 3. 학습 보고서**

1. **자신이 담당한 역할**
   * + 모델 조사
     + 데이터 전처리
     + 결과물 시각화 및 통계 분석
2. **모델 선택 및 그 이유**
   * + ELECTRA는 NLP STS task에서 SOTA의 성능을 보이는 모델 중 하나이다.
     + 그 중에서 KoELECTRA는 한국어 Pretrained Model 중 유사도 분석 TASK에서 가장 높은 성능을 보인 검증된 모델 중 하나이다.

테이블이(가) 표시된 사진

자동 생성된 설명

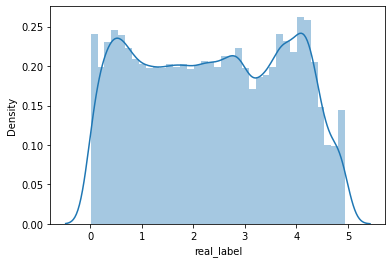
* + - 유사도 측정에 강세를 보이는 SentenceBERT와 성능을 비교 시험해 본 결과, pearson score가 88:92 로 더 높은 성능임을 확인하였다.
    - Replaced token detection을 활용하는 ELECTRA 모델은 계산량이 기존 BERT 계열 모델보다 적고 효율적이다.(속도 면에서 우세)

1. **파라미터 튜닝 및 결과**

**Wandb 활용하여 최적의 parameter 선정**

* + epochs: 1
  + grad\_norm: 1
  + learning\_rate: 5e-05
  + max\_length: 128
  + train\_batch\_size: 32
  + warm\_up\_ratio: 0
  + weight\_decay: 0.01

1. **훈련 과정**
   1. 전처리 : 홈페이지 주소 및 특수문자 등 불필요한 부분을 전처리하였다.
   2. KOELECTRA로 Training& Validation 진행
      1. Metric은 validation loss로 정하여 학습을 진행하였다.
      2. Validation loss는 떨어지지 않지만, Metric인 F1-Score 및 Pearson’s r 은 상승하는 현상을 확인하고 개선의 필요성을 확인하였다.
   3. Label 불균형 해소를 위한 Data Augumentation
      1. Test set과 validation set의 분포가 균일하지 않다면, test set의 recall과 precision의 편차가 크고 결과적으로 F1score가 더 낮게 나타날 것이라고 판단하였다.
      2. Test set의 분포를 미리 측정할 수는 없으나, 1~5까지의 Label이 균일하도록 Augumentation 진행
      3. [Augumented SBERT](https://www.sbert.net/examples/training/data_augmentation/README.html) : SentenceBERT에서 Data augumentation을 진행했던 논문을 참고하여 문장 pair와 label(유사도)을 또다른 모델로부터 생성 (기존 Gold Dataset + 새로 만든 Silver Dataset)
      4. Label 생성을 위해 semantic search 로 sampling하였으며, 논문에는 bert를 사용하였으나, 성능이 더 좋을 것으로 판단되는 Roberta-large를 사용하였다.



[Data 생성 후 균일해진 Training label 분포]

1. **API 서버 코드[**[**Repository**](https://github.com/zzhenxi/Semantic-Textual-Similarity-API)**]**
   1. 모듈화 및 로컬호스팅으로 유사도를 확인할 수 있도록 구성

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. 디렉터리 구조

텍스트이(가) 표시된 사진

자동 생성된 설명

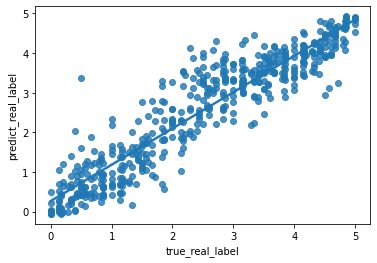
* + 1. running\_model : fine\_tuning 된 best model과 필요한 모듈을 포함
       1. config.json : hyperparmeter 설정값 포함
       2. pytorch\_model : model bin 파일
       3. data\_preprocessing.py : 전처리 모듈
    2. templates : 화면 UI 구성을 위한 HTML 템플릿
    3. main.py : Flask를 이용한 REST API 구현

1. **최종 결과 분석**

텍스트, 실내, 스크린샷이(가) 표시된 사진

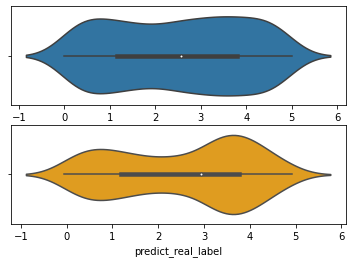
자동 생성된 설명

* 1. **최종 결과 및 성능**
     1. Binary Classification(F1 : 0.8668)
     + precision : 0.81 /recall : 0.93
     + recall>> precision의 의미
       1. 유사하지 않은데 유사하다고 예측한 것(fp)이 유사한데 유사하지 않다고 예측한 것(fn)보다 많다.
       2. 모델은 대체로 실제보다 유사하다고 예측하는 비율이 높다.
       3. threshold 값을 조금씩 낮추면서 성능을 비교하면 더 높은 f1 score를 기대해 볼 수 있다.
     1. Regression (Pearson’s r : 0.933)
        1. True label VS predicted label



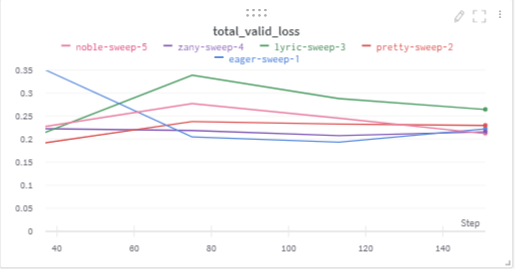
대체로 상관계수에 맞는 양의 상관관계 분포를 확인할 수 있었다.

* + - 1. 분포 비교



실제로는 균일한 반면, 예측한 유사도 분포의 경우 비교적 더 3,4 쪽에 치우쳐져 있는 것을 확인할 수 있었다. 이는 precision보다 recall이 높은 결과와 일맥상통하는 부분임을 확인할 수 있다.

* 1. 고찰
     1. 전처리로 인해 F1 score 0.7, correlation coefficient 0.1 정도의 성능 향상이 있었다.
     2. Data Augumentation은 Validation loss를 더 감소시켜주지만, Test set에서의 성능을 늘려주지는 못했다. 이는 새로 생성한 데이터의 labelling 성능과 실제 label간의 gap 으로부터 비롯된 것으로 사료된다.

[Augumentation 후 개선된 valid loss 확인]

* + 1. F1 score는 Threshold 값을 3보다 조금 더 높은 값으로 올려서 classification을 진행하면 classification의 성능을 확인할 수 있을 것이라 예상된다.
    2. 향후 database 구축 및 sqlalchemy 활용한 연결, 모듈 고도화를 통해 프로젝트를 발전시켜나갈 수 있을 것으로 보인다.