기업과제3_3팀_보고서

1. Abstract

본 프로젝트에서는 주어진 두 문장에 대해 의미적 유사도를 예측하는 모델을 만들고자 하였다.

모델을 학습하는 데에는 두 개의 한국어 문장 쌍들에 대해 유사도를 측정한 KLUE-STS(Semantic Textual Similarity) 데이터가 이용되었다. 해당 데이터는 airbnb, policy, paraKQC 의 도메인에서 추출된 문장들의 유사도를 0~5 사이의 값으로 기록('real-label')했다 (Human-labeled data).

실험을 위해 해당 benchmark에서 SOTA 성능을 보였던 KLUE-BERT-base, KLUE-RoBERTa-base, KoELECTRA-base-v3-discriminator 의 Pre-trained 모델들을 채택하였다. 뿐만 아니라 우수한 문장 임베딩들을 생성하여 빠르게 유사도를 산출해낼 수 있는 모델인 Sentence Transformer 를 이용하였다.

예측한 유사도의 성능을 평가하고자 Pearson's correlation coefficient 와 F1 score 를 평가지표로 채택했다. 이에 <KLUE: Korean Language Understanding Evaluation>(2021) 논문에서 구현된 방식과 마찬가지로 STS task 를 sentence pair regression task 로 정의하여, 각 문장 쌍 간의 의미적 유사도를 계산하고 모든 데이터에 대한 예측 값과 실제 값을 비교하여 Pearson's correlation coefficient 를 측정하였다. F1 score 는 threshold 를 3.0 으로 설정하여 해당 값 이상은 1, 미만은 0 으로 labeling 을 진행했다. 이를 통해 전체 데이터에 대한 binary prediction 값들과 실제 binary label 을 비교하여 F1 score 를 산출하였다. 이 과정에서 주어진 데이터의 유사도와 다른 개념/범위의 유사도, 즉 두 문장 간의 cosine similarity 를 측정하는 Sentence Transformers 를 추가 학습 과정에서 제외하였다.

다른 모델들을 본격적으로 Fine-tuning 하기에 앞서, 탐색적 데이터 분석(EDA)를 통해 중복 데이터, 문장 길이, 도메인 분포, 유사도 분포를 확인했다. 이를 바탕으로 중복 데이터를 제거하고, 불균형한 유사도 분포 문제를 해결하고자 BM25 Sampling 방법을 이용하여 데이터를 증강하였다. 결과적으로 도메인 분포를 고려하지 못한 데이터 증강은 성능 향상을 도모하지 못했다.

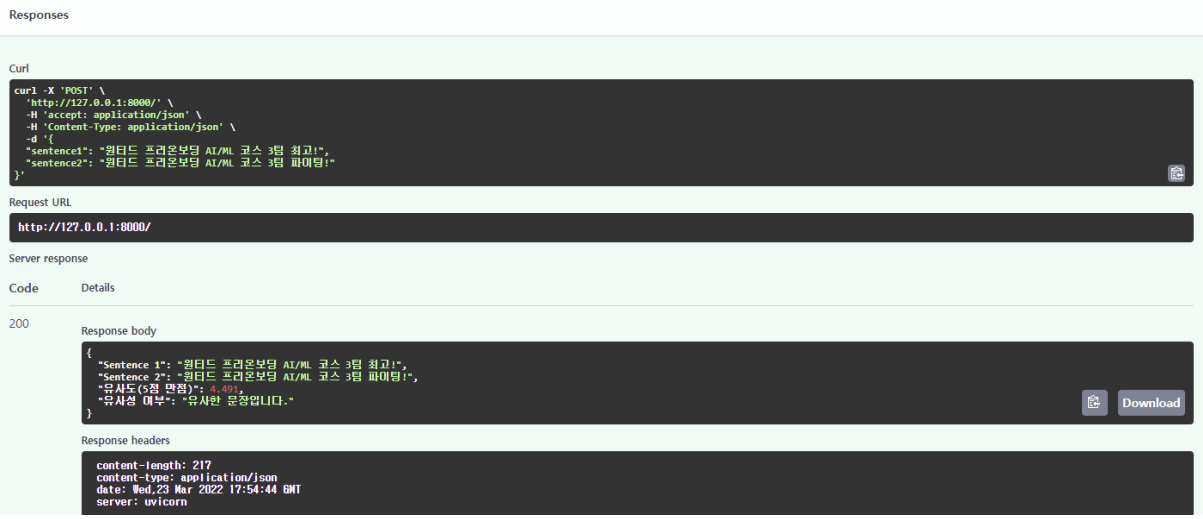
Tuning 하고자 하는 Hyper-parameter 로는 모델의 학습률을 의미하는 learning rate 와 regularization 효과를 통해 overfitting 을 방지할 수 있는 weight decay 를 선정하였다. Optuna 를 통해 해당 hyper-parameter 들에 대한 tuning 을 진행했으며 WandB 를 통해 학습 과정과 결과 값을 관찰했다.

결과적으로 KLUE-RoBERTa-base와 KoELECTRA-base-v3-discriminator는 비슷한 성능을 보였으며 KLUE-BERT-base는 두 모델들보다 Pearson'r 과 F1 score 에서 모두 0.03 정도 낮은 성능을 갖는 것으로 확인되었다. STS task 를 sentence pair regression task 로 정의했기 때문에, 더 높은 Pearson's correlation coefficient 를 기록한 KLUE-RoBERTa-base를 최종 모델로 채택하였다.

그리고 최종 모델을 바탕으로 두 한국어 문장의 의미적 유사도를 추론하는 Web Demo Page 를 만들기 위해 FastAPI 로 다음과 같은 서비스를 구현하였다.

Demo Link:

https://github.com/SYKflyingintheSKY/Wanted_PreOnBoarding_AI/tree/main/기업과제/과제3_STS



2. 모델 선정

KLUE STS benchmark 에서 좋은 성능을 보였던 pre-trained model 들을 선택하였다.

2.1. KLUE-BERT-base

- https://huggingface.co/klue/bert-base
- BERT (Bidirectional Encoder Representations from Transformers)

    - Masked Language Modeling (MLM), Next Sentence Prediction (NSP) 의 objective로 학습된 Pre-trained 모델이다.
    - 양방향으로 학습을 진행하는 Bi-directional Language Model 이다. 때문에 NLU(Natural Language Understanding) 에서 좋은 성능을 기록했다.
- KLUE(Korean Language Understanding Evaluation) 데이터를 이용해 사전 학습을 진행한 모델이다.

2.2. KLUE-RoBERTa-base

- https://huggingface.co/klue/roberta-base
- RoBERTa (A Robustly Optimized BERT Pretraining Approach)
    - BERT 보다 더 많은 데이터(x 10), 더 긴 sequence, 더 큰 batch size, 더 긴 training period 를 통해 만들어진 Pre-trained 모델이다.
    - 성능이 다소 낮았던 NSP objective를 삭제하고 MLM task 로 사전 학습을 진행하였다.
    - Dynamic masking 기법을 이용하여 batch 마다 masking 을 새로 적용해줌으로써 다양성을 확보했다.
- KLUE(Korean Language Understanding Evaluation) 데이터를 이용해 사전 학습을 진행한 모델이다.

2.3. KoELECTRA-base-v3-discriminator

- https://github.com/monologg/KoELECTRA
- ELECTRA(Efficiently Learning an Encoder that Classifies Token Replacements Accurately)
    - Replaced Token Detection (RTD) 로 사전 학습을 진행했다.
        - generator를 통해 실제 input 과 비슷한 token을 생성하여, 각 토큰이 original token인지 판단함으로써 이진 분류 문제로 치환하여 효율성을 확보했다.
        - 입력의 15% 가 아닌 모든 input token에 대해 학습하였다.
- v3 의 경우 약 34G의 한국어 Corpus 로 사전 학습을 진행했다.

2.4. Sentence Transformers

- https://huggingface.co/sentence-transformers
- Sentence Transformers
    - 사전 학습된 BERT, RoBERTa, ELECTRA 와 결합하여 사용 가능한 구조이다. e.g. SBERT, SRoBERTa, SELECTRA
    - Siamese Network 등을 활용하여 모든 문장을 단일 Encoder 형식으로 연산하기 때문에 각 문장의 embedding 계산을 독립적으로 수행할 수 있다.
    - 이를 통해 우수한 sentence embedding 을 얻을 수 있다.
    - 검색 후보 문서 등에 대해 미리 embedding 계산이 가능하다. 때문에 새로운 search query 의 embedding 만 추가적으로 산출한 뒤 유사도를 계산하는 방식으로 속도를 개선했다.

아래의 표에서 각 모델의 크기와 pre-trained data. KLUE-STS Benchmark 에 대한 Baseline Score 를 확인할 수 있다.

Model	Model SIze	Pretrained Data	Baseline Score (KLUE-STS)
KLUE-BERT-base	- Layers: 12 - Hidden Size: 768 - Embedding Size: 768 - Head: 12	KLUE	- Pearson's: 92.50 - F1 score: 85.40
KLUE-RoBERTa-base	- Layers: 12 - Hidden Size: 768 - Embedding Size: 768 - Head: 12	KLUE	- Pearson's: 92.50 - F1 score: 85.40
KoELECTRA-base-v3-discriminator	- Layers: 12 - Hidden Size: 768 - Embedding Size: 768 - Head: 12	Korean News, Wiki, 나무위키, 모두의 말뭉치 (신문, 문어, 구어, 메신저, 웹)	- Pearson's: 92.46 - F1 score: 84.84
Sentence Transformers	같이 사용되는 pre-trained model 의 model size 를 따른다.	같이 사용되는 모델의 사전 학습 데이터	-

3. 데이터 전처리

- train data의 중복을 제거했다 (11668개 → 11661개)
- 대부분 문장 당 단어 개수가 30개 이하이므로 문장 길이에 대한 추가 전처리 작업은 진행하지 않았다.
- Data Augmentation: BM25 Sampling
  - BM25 Sampling 기법을 사용한 Augmented SBERT 가 좋은 성능을 보였다는 점에 기인하여 주어진 STS 데이터를 증강시켰다.
  - BM25 Sampling
    - 주어진 쿼리에 대해 문서와의 연관성을 평가하는 랭킹 함수 알고리즘이다.
    - 프로세스
      - ElasticSearch 3로 모든 고유 문장을 검색 쿼리용 색인화로 적용하였다.
      - 각 문장에 대한 상위 3개의 유사한 문장 검색하고, 이들 쌍을 인코더로 labeling 하였다.
      - 새로 생성된 모든 pair 를 Silver Dataset으로 이용했다.
  - BM25 Sampling 을 통해 생성된 Silver Dataset 과 주어진 Gold Dataset (KLUE-STS) 을 이용하여 훈련을 진행하였다. Gold Dataset 의 real-label distribution 의 균형을 맞추기 위해, [0, 1), [1, 2), [2, 3), [3, 4), [4, 5] 범위 내 데이터 수를 같도록 (각 5000개) 증강시켰다.
  - 결과: 기존 KLUE-STS 데이터만을 사용할 때보다 낮은 성능을 기록했다. Source Domain Balance (airbnb, policy, paraKQC) 를 고려하지 못한 데이터 증강이라는 점과 최적화 되지 않은 Cross Encoder 모델을 통해 증강된 데이터의 label 을 채워 넣은 점이 한계로 작용한 것이라 생각된다.
- train data 를 train : validation = 9:1 의 비율로 나누어 실험을 진행하였다.

Dataset	Size
Train	10494
Validation	1167
Test	519

4. 훈련 및 평가 과정

- <KLUE: Korean Language Understanding Evaluation>(2021) 에서 STS task 를 수행한 방식으로 BERT, RoBERTa, ELECTRA 를 이용하여 훈련 및 평가를 진행했다.
  - Sentence Pair Regression Task: 두 input sentences 의 의미적 유사도를 STS Dataset 의 ‘real-label’을 이용하여 regression 을 통해 측정했다.
    - 의미적 유사도 = human-labeled data = 범위: 0 (겹치는 의미 없음) ~ 5 (동등한 의미)
    - 평가 지표: Pearson’s correlation coefficient = human-labeled 문장 유사도 점수와 모델 예측 점수 간의 선형 상관 관계를 측정하는 지표이다.
  - Binary classification Task: Sentence Pair Regression Task 를 통해 측정된 값을 3.0 의 threshold 를 기준으로 0과 1로 분류하였다.
    - 0: 두 문장이 유사하지 않음 (not paraphrased), 1: 두 문장이 유사함 (paraphrased)
    - 평가 지표: F1 score = 실제 STS Dataset 의 ‘binary-label’과 모델의 binary predictions 에 대한 precision 과 recall 에 대한 조화 평균을 측정했다.
  - 이러한 이유로 모델의 주 Task 를 regression 으로 잡고 MSE Loss 를 최소화 하는 방식으로 모델을 학습했다. 다만 정규성 가정을 만족할 때 Cross Entropy 를 최소화 하는 과정이 L2 loss 를 최소화하는 과정과 비슷하다는 점을 이용해 이 또한 regression 의 cost function 으로 이용해 보았다. 주어진 data의 real-label distribution 이 완벽한 정규 분포를 따르지는 않았지만, [0,1] 구간을 제외하고는 어느 정도 정규성을 띄었기 때문에 해당 실험을 시도해보았다. 결과적으로 Cross Entropy 를 loss function 으로 사용했을 때 0.03 정도의 성능 향상을 기록했다.

- Sentence Transformer 의 한계점
  - Sentence Transformer 는 두 문장 쌍에 대한 cosine similarity score 를 계산한다.
    - human-labeled 유사도 점수에 기초한 것이 아닌 다른 개념의 유사도를 산출한다.
    - 더불어 cosine similarity 는 -1 ~ 1 사이의 값을 갖는 반면, KLUE-STS 의 real-label은 0 ~ 5 사이의 값을 갖는다.
    - 때문에 classification 을 위한 threshold 를 정할 수 있는 기준 또한 모호하다.

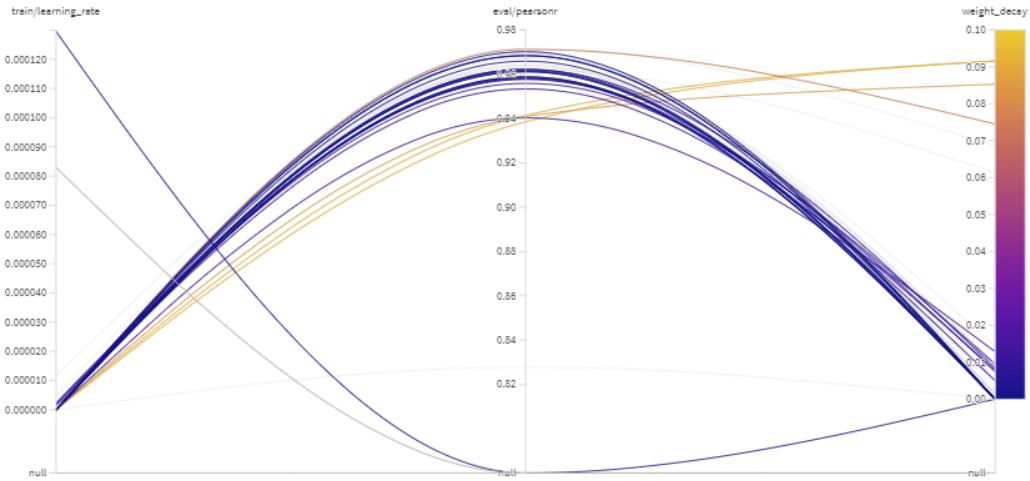
- 따라서 Sentence Transformer 가 예측한 유사도와 STS 의 예측값 사이에서 산출된 Pearson’r 과 F1 score 의 유효성에 대한 의문이 제기되었다.
- 또한 속도는 빠르지만 일반적인 transformers 모델들보다 낮은 성능을 보였다.
- 이러한 이유들로 인해 추가적인 hyper-parameter tuning 과정을 거치지 않고 학습을 중단했다..

5. Hyper-Parameter Tuning

- Optuna 를 이용하여 hyper-parameter tuning 을 진행하고, WandB를 이용해 결과를 분석했다.
- learning rate 와 weight decay 에 대한 hyper-parameter tuning 을 진행하였다.
  - learning rate: 모델의 학습률에 의한 성능을 확인하고자 했다.
  - weight decay: regularization 의 효과를 보기 위해 선택하였다.
- 아래의 그림
  - 왼쪽 axis: training learning rate
  - 오른쪽 axis: weight decay
  - 가운데: validation set 의 Pearson’r score

  learning rate는 1e-6 ~ 1e-5, weight decay는 0 ~ 0.15 사이의 값일 때 높은 성능을 보였다.

  따라서 최종 모델은 이 사이의 값들로 learning rate 와 weight decay를 정하여 학습을 진행했다.



6. 최종 결과 분석

- 각 모델 당 가장 높았던 성능은 아래의 표에서 확인할 수 있다.
- 결과적으로, KLUE-RoBERTa-base ≥ KoELECTRA-base-v3-discriminator > KLUE-BERT-base 의 순으로 높은 점수를 기록했다.
- KLUE-RoBERTa-base 와 KoELECTRA-base-v3-discriminator는 비슷한 성능을 보였지만 1차적인 task 는 regression 이었기 때문에 Pearson’s correlation coefficient 가 더 높은 KLUE-RoBERTa-base 를 최종 모델로 채택하였다.
- KLUE-BERT-base는 다른 두 모델보다 Pearson’r 과 F1 score 모두에서 0.03 정도 낮은 성능을 보였다:
    - vs. KLUE-RoBERTa-base: optimized 된 training 과정(학습 데이터 양, 시간, 문장 길이, 배치사이즈) 뿐만 아니라 dynamic masking 기법을 통해 다양한 토큰을 고려한 점이 유사도를 예측하는데 긍정적인 영향을 미쳤을 것으로 해석된다.
    - vs. KoELECTRA-base-v3-discriminator: STS task를 수행함에 있어 일부 토큰보다 모든 토큰을 고려하는 방법이 효과적임을 파악하였다.

Pre-trained Model	Test Score	Epoch	Batch Size	Optimizer	Learning Rate	Scheduler	Warmup Steps
KLUE-BERT-base	Pearson's: 0.8984 F1score: 0.8214	4	32	AdamW	2e-5	Linear Warmup	132
KLUE-RoBERTa-base	Pearson's: 0.9266 F1score: 0.8589	4	32	AdamW	2e-5	Linear Warmup	132
KoELECTRA-base-v3-discriminator	Pearson's: 0.9185 F1score: 0.8675	4	32	AdamW	2e-5	Linear Warmup	132

7. 한계 및 개선 방안

- 한계: 도메인을 고려하지 못한 Data Augmentation 진행으로 성능 향상을 꾀하지 못했다.
- 개선 방안: 도메인까지 고려할 수 있는 sub-sampling 방법을 고안하여 적용할 수 있을 것으로 기대된다.