

Semantic Text Similarity (STS)

NLP-09 (Time Flies)

김인수, 오주영, 양서현, 문지원, 손윤환

1. Intro

1.1. 개요

- **Semantic Text Similarity**: 복수의 문장에 대한 유사도를 선형적 수치로 제시하는 NLP Task.
- 이는 두 문장이 서로 동등하다는 양방향성 가정하고 진행됨.
- 이러한 수치화 가능한 양방향 동등성은 정보 추출, 질문-답변 및 요약과 같은 NLP 작업 전반에 널리 활용 및 응용.
- 대회 목표는 STS 데이터셋을 활용해 두 문장의 유사도를 측정하는 AI 모델의 구축.
 - [0, 5] 범위의 유사도 점수를 출력

• Label 점수: 0 ~ 5사이의 실수

- 5점: 두 문장의 핵심 내용이 동일하며, 부가적인 내용들도 동일함
- 4점: 두 문장의 핵심 내용이 동등하며, 부가적인 내용에서는 미미한 차이가 있음
- 3점: 두 문장의 핵심 내용은 대략적으로 동등하지만, 부가적인 내용에 무시하기 어려운 차이가 있음
- 2점: 두 문장의 핵심 내용은 동등하지 않지만, 몇 가지 부가적인 내용을 공유함
- 1점: 두 문장의 핵심 내용은 동등하지 않지만, 비슷한 주제를 다루고 있음
- 0점: 두 문장의 핵심 내용이 동등하지 않고, 부가적인 내용에서도 공통점이 없음

- 학습 데이터셋 9,234개, 검증 데이터셋 550개, 평가 데이터셋 1,100개.
 - 평가 데이터의 50%는 Public 점수 계산에 활용, 실시간 리더보드에 반영되며 남은 50%는 Private 결과 계산에 활용되어 최종 평가에 반영.
- 최종 결과물은 .csv 형태로 제출.
 - 입력: 두 개의 문장과 ID, 유사도 정보
 - 출력: 평가 데이터에 있는 각 문장쌍에 대한 ID와 유사도 점수

id	source	sentence_1	sentence_2	label	binary-label
boostcamp-sts-v1-train-000	namc-sampled	스킬도있고 반전도 있고 예는 한국영화 소개기를하고는 차원이 다르네요~	반전도 있고,스킬도 있고제미도있네요.	2.2	0.0
boostcamp-sts-v1-train-001	slack-rtl	웃 제가 접근권한이 없다고 합니다;	오, 액세스 권한이 없다고 합니다.	4.2	1.0
boostcamp-sts-v1-train-002	petition-sampled	주제참목조건 변경해주세요.	주제참목 무주제기준 변경해주세요.	2.4	0.0
boostcamp-sts-v1-train-003	slack-sampled	일시후 처음 대면으로 만나 반가웠습니다.	확실오로만 보다가 리얼로 만나니 정말 반가웠습니다.	3.0	1.0
boostcamp-sts-v1-train-004	slack-sampled	후도후도 하네요	포록 실제로 한번 봐요 후후후~	0.0	0.0
boostcamp-sts-v1-train-005	namc-rtl	오마이갓지저스크라이스트를	오 마이 갓 지저스 스크론 이스트 램	2.6	1.0
boostcamp-sts-v1-train-006	slack-rtl	전 알만 찍어도 러만 하능.. wrw	알만 찍어도 하능은 러말다.. wrw	3.6	1.0
boostcamp-sts-v1-train-007	namc-sampled	이렇게 귀여운 귀들은 처음아네요. * * *	이렇게 지겨운 귀로영화는 처음..	0.6	0.0
boostcamp-sts-v1-train-008	petition-sampled	미세먼지 해결이 가장 시급한 문제입니다	가장 시급한 것이 산생아실 관리입니다	0.4	0.0
boostcamp-sts-v1-train-009	petition-sampled	크림하우스 환불조치해주세요.	크림하우스 환불조치할 수 있도록해주세요	4.2	1.0
boostcamp-sts-v1-train-010	slack-rtl	그 뭐부터 연봉 커내줘야 겠어요!	책에서 커내야겠어요!	2.4	0.0

- 평가 기준은 예측과 정답 간의 Pearson Correlation Coefficient으로 삼는다.
- 개별 예측의 일치보다는 전체적인 경향의 유사도가 중요.

$$\text{피어슨 상관계수} = \frac{\text{공분산}}{\text{표준편차} \cdot \text{표준편차}}$$



- 프로젝트 전체 기간 (2주) : 12월 11일 (월) 10:00 ~ 12월 21일 (목) 19:00
 - 팀 병합 기간 : 12월 12일 (화) 16:00 까지
 - 팀명 컨벤션 : 도메인_팀번호(2자리)조 / ex) CV_03조, NLP_02조, RecSys_08조
 - 리더보드 제출 오픈 : 12월 13일 (수) 10:00
 - 리더보드 제출 마감 : 12월 21일 (목) 19:00
 - 최종 리더보드 (Private) 공개 : 12월 21일 (목) 20:00
- GPU 서버 할당 : 12월 13일 (수) 10:00
- GPU 서버 회수 : 12월 29일 (금) 16:00

1.2. 프로젝트 구조

```

└code.tar.gz
├┐
├┐  train.py
├┐    # 데이터 학습에 사용되는 코드
├┐  inference.py
├┐    # 학습된 모델로 데이터를 예측하는 코드
├┐  requirements.txt
├┐    # train.py와 inference.py를 실행하는데 필요한 라이브러리
├┐
└┐
  
```

- Train part의 순서는 다음과 같습니다:
 1. Dataloader와 Dataset 클래스를 사용해 데이터를 준비합니다.
 1. 사용하는 선학습 모델에 맞는 토큰나이징을 활용하여, 두 문장을 [SEP] 토큰으로 이어붙여서 활용합니다.
 2. 토큰의 최대 길이는 160으로 사용하고, 토큰이 부족하다면 [PAD] 토큰을 추가하고, 토큰이 넘친다면 뒷 부분을 잘라 활용합니다.
 2. Model 클래스를 사용해 모델을 설정하고 준비합니다.
 3. Trainer와 Dataloader, Model을 사용하여 학습을 수행합니다.
 4. 학습이 완료되면, `torch.save(model, 'model.pt')` 를 통해 학습된 모델을 model.pt라는 파일로 저장하게 됩니다.

- Inference part의 순서는 다음과 같습니다 :
 1. train.py에서 저장된 model.pt를 호출하여 데이터 예측을 진행합니다.
 2. 예측된 결과를 제출 형식에 맞게 변경하여 준비합니다.
 3. sample_submission.csv를 불러와 target만 예측된 결과로 변경하고 output.csv로 저장합니다.
- 코드 실행이 끝나면 output.csv라는 예측 결과 파일이 생성되고 리더보드에 제출할 수 있습니다.

1.3. 프로젝트 환경

- GPU: V100 * 5, 각자의 IDE에 SSH 연결하여 작업 수행
- 협업 관리: [Notion](#), [Github](#)

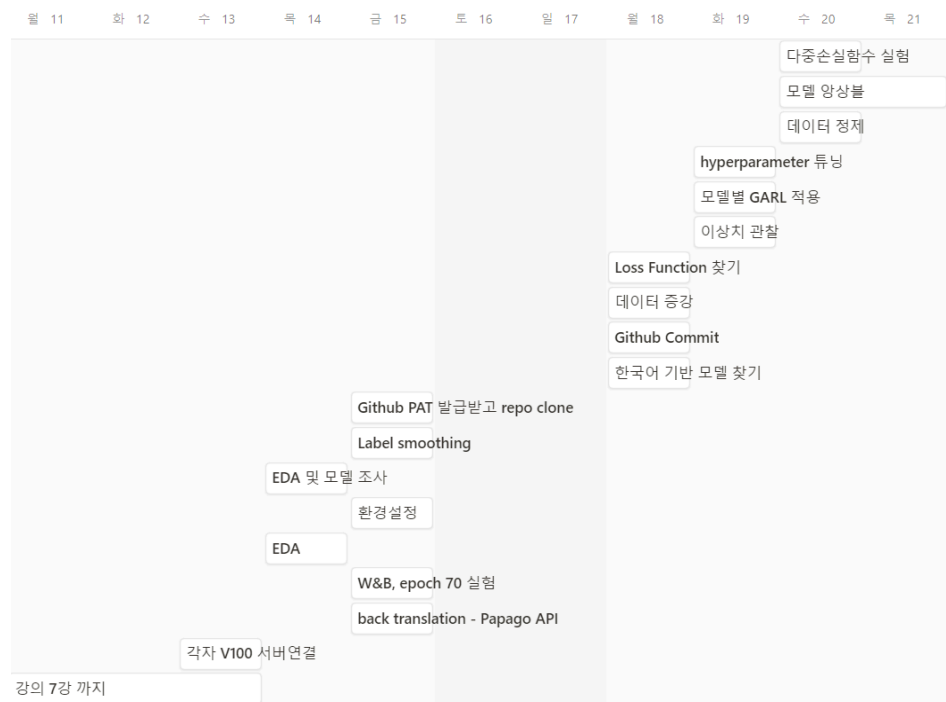
2. 프로젝트 팀 구성 및 역할

팀원	역할
공통	데이터 분석, 실험 수립 및 진행, 하이퍼 파라미터 튜닝, Wrap Up 리포트 작성
김인수	앙상블 베이스라인 코드 작성, 데이터 증강(bert)
오주영	손실함수 탐색, 다양한 손실함수 비교/분석 및 최적화
양서현	train-dev 데이터 분석, 데이터 증강(Label Smoothing, Copied sentence, 문장 교정)
문지원	한국어 기반 모델 탐색, 데이터 전처리 실험(특수문자 제거, 형태소 분석)
손윤환	한국어 기반 모델 탐색, 예측 데이터 이상치 분석, 데이터 증강 실험(역번역)

3. 프로젝트 수행 절차 및 방법

3.1. 진행 절차

- 1) 강의 수강 및 사전 학습
- 2) 데이터 EDA, 다양한 접근 방법 조사 및 실험
- 3) 진행한 실험 공유 및 유의미한 접근법 선정
- 4) 선정한 접근법 기반 역할 분배 및 실험 진행
- 5) 모델 튜닝 및 앙상블 통해 최종 결과물 도출



3.2. 협업 문화

- 1) 팀 일일계획표를 작성하며 각자 목표 및 진행 상황 공유함
- 2) 자료 및 정보 공유를 위해 공용 노션 페이지를 활용함
- 3) 질문 또는 어려운 부분이 있을 시 실시간 회의 때 함께 보며 해결함

4. 프로젝트 수행 결과

4-1. EDA (Exploratory Data Analysis)

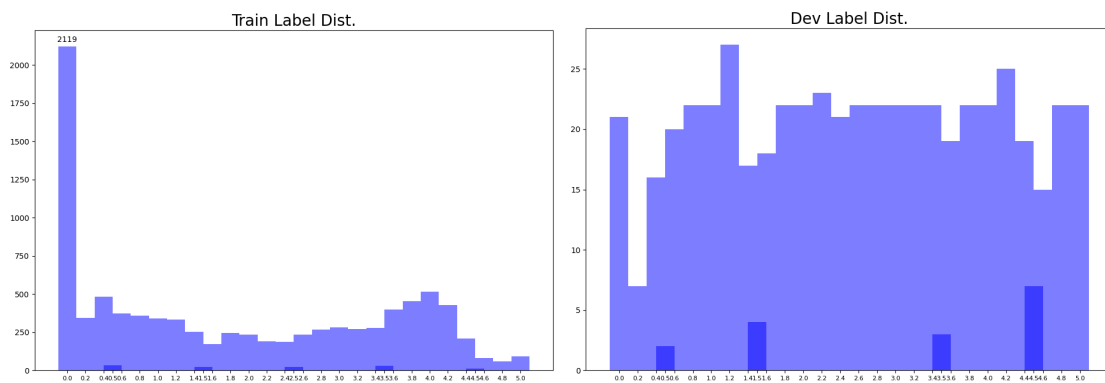
주어진 데이터의 label 분포, source 분포, 문장 길이, 문장 형태 분석과 관련된 탐색적 데이터 분석을 진행하였고, 이를 통해 성능 개선 가설, 전략을 수립하였다.

A. Basic Data Information

Train data의 개수는 총 9,324개, dev data의 개수는 총 550개 그리고 test data의 개수는 총 1100개이며, 3개의 dataset 모두 null값은 존재하지 않았다. target column label은 0~5 사이 float64 type으로 해당 task는 Regression 문제에 해당한다고 보았다.

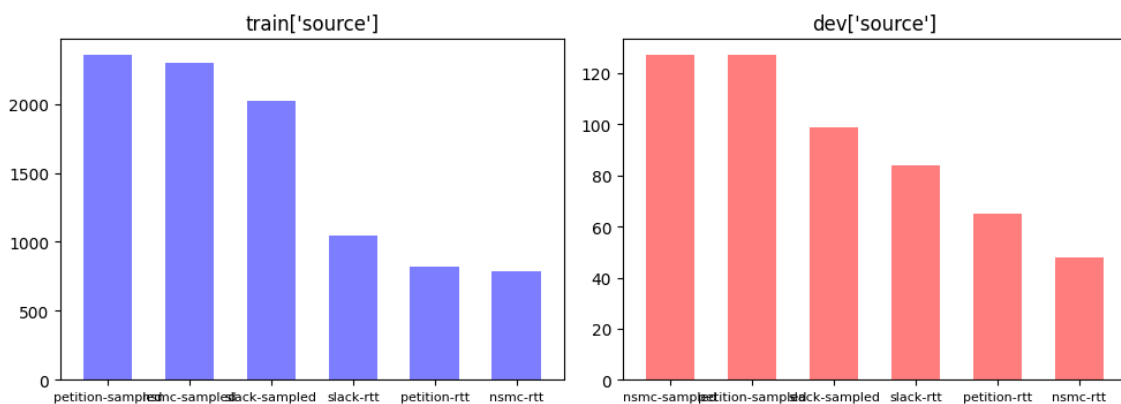
B. Label 분포

train와 dev data의 분포를 barplot 시각화를 통해 확인했다. train data은 0에 편향돼 있고, dev data는 골고루 분포되어 있다. raw train data로 학습 시에 label 값 편향이 있어 모델의 성능이 낮아질 수 있는 가능성을 알 수 있었다.



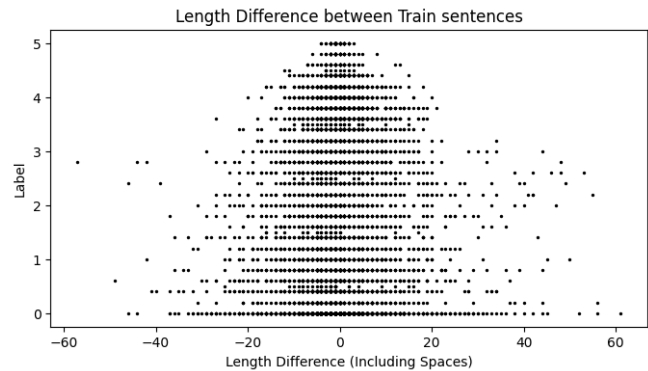
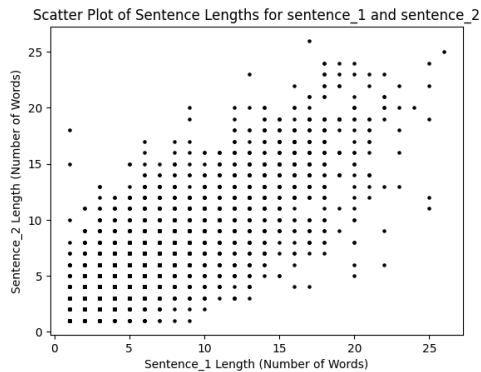
C. Source 분포

train과 dev data의 분포는 비슷하다. one-hot encoding을 이용해 Object형을 숫자형으로 변환 이용 가능성을 제시할 수 있었다.



D. 문장 길이 분석

train data의 두 문장 길이는 비슷하나, 이상치의 존재에 대해 결과의 편향이 발생할 수 있음을 인지할 수 있었다. dev data와의 분포도 비슷하였다.



E. 문장 형태 분석

- 특수문자 - <Person>, [UNK], !!!, ^^ 등 마스킹, 특수문자의 중복
- 반복되는 한글 자모음 - ㅋㅋㅋㅋ/ㅎㅎ/ㅋㅋ 등 자음, 모음이 3개 이상 반복되는 경우
- 맞춤법, 띄어쓰기 - 맞춤법과 띄어쓰기가 맞지 않은 문장 다수

4-2. 실험 방법 및 과정

A. 모델 측면

kykim/electra-kor-base

새벽에 보다가 이유모를 [UNK], 보면서 계속 [UNK], 0.9, 3.0

기존 klue/roberta-base 모델 학습 당시 valid 예측 결과를 분석한 결과 [UNK] 토큰으로 인해 예측이 제대로 이루어지지 않는 상황 발생 -> tokenizer vocab 크기가 큰 모델을 탐색
기존 32K 사이즈보다 큰 42K word piece 알고리즘 기반 학습모델 kykim/bert 모델 중
KorSTS 벤치마크 성능이 가장 뛰어난 electra 모델 사용

snunlp/KR-ELECTRA-discriminator

KorSTS 벤치마크 기준으로 상위권 모델

Mecab-Ko 형태소 분석기 기반 사전 -> 한국어 기반으로 구축된 양질의 사전을 통해 성능 향상 기대

monologg/koelectra-base-v3-discriminator

많이 사용되는 여러 한국어 기반 모델과 현재 프로젝트에서 선택한 모델에 비교해서 KorSTS 벤치마크를 기준으로 더 뛰어난 성능의 모델 추가

Ensemble

(**kykim(증강), snu, snu(증강), monologg, snu(aug+hanspell)**)

모델의 일반화 및 성능 향상을 위해 앙상블 Weighted sum 모델 적용

Test pearson 기준 0.92가 넘는 모델 후보군 설정. Inference의 다양성 확보를 위해 증강

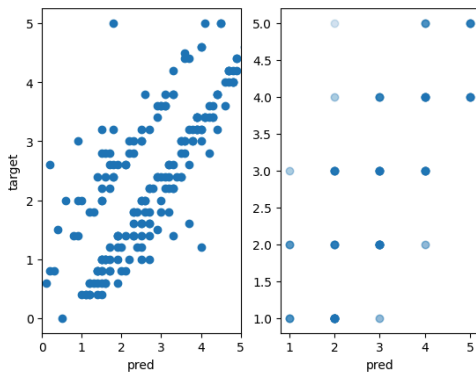
데이터로 학습한 모델과 아닌 모델들을 앙상블(예측 결과 모든 라벨을 고르게 예측)

B. 손실함수 측면

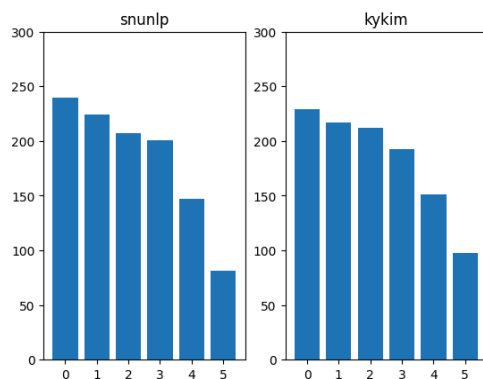
- 손실함수의 설정이 모델의 학습 방향에 큰 영향을 미치기 때문에 적절한 손실함수 사용이 성능에 영향을 줄 것이라는 가정에서 다양한 손실함수를 적용해보기로 결정.
- 실험에 적용한 손실함수 : SmoothL1Loss, MSELoss, pearson score, General and Adaptive Robust Loss Function, MSE + Huber, MSE + noise(CosineSimilarity), MSE+pearson
- 각 손실함수의 설정 의의와 결과는 노선의 STS Project - 실험결과 참조.

C. 데이터 측면

Inference output 결과 분석



- valid 예측-정답 간의 점수 차이 분포를 비교, 분석 결과 고득점 label에 대해 낮은 예측값을 출력하는 경우가 다수 발생함을 확인
- [UNK] 토큰과 문장 의미상 오타가 발생하는 경우 예측이 제대로 이루어지지 않음
- Py-Haspell 라이브러리를 이용한 오타 교정 및 tokenizer 개선 모델 탐색 진행



- 각 모델 별 valid 예측 분포를 분석하여 향후 앙상블 과정에서 weighted sum 계산 기준 수립

Preprocessing Data 및 결과

A. 데이터 정제(data cleaning)

- 맞춤법 검사(Hanspell)

맞춤법 검사 라이브러리 Py-Hanspell 내 `spell_checker`를 활용했다. 맞춤법, 띄어쓰기 오류가 많이 개선하였으며, [UNK] 등 마스킹을 삭제하여 실험시 성능이 향상되었다.

- regexp

정규식을 활용해 특수문자를 제거했다. `regexp`만 사용하여 실험 시 `raw`와 성능이 비슷하지만 다른 전처리 기법과 같이 사용했을 때 성능이 떨어져 채택하지 않았다.

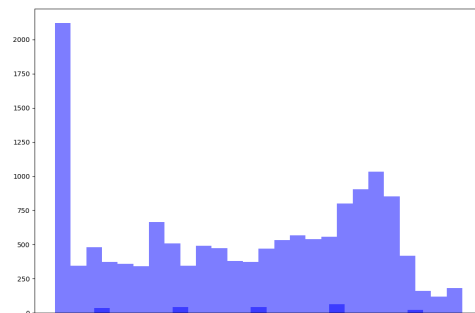
B. 데이터 증강(data augmentation)

- Bert Augmentation(<https://github.com/kyle-bong/K-TACC>)

BERT 모델을 통해 문맥을 고려하여 [MASK] 토큰을 복원하는 방식으로 데이터를 증강하는 방법. Insertion 방법과 Replacement 2가지 방법 중 원본 문장의 단어는 그대로 보존한 채 단어나 기호를 추가하는 방식인 Replacement 방법 채택.

실험 결과 중 가장 높은 Pearson's correlation(0.9300)

- label smoothing, Copied sentence

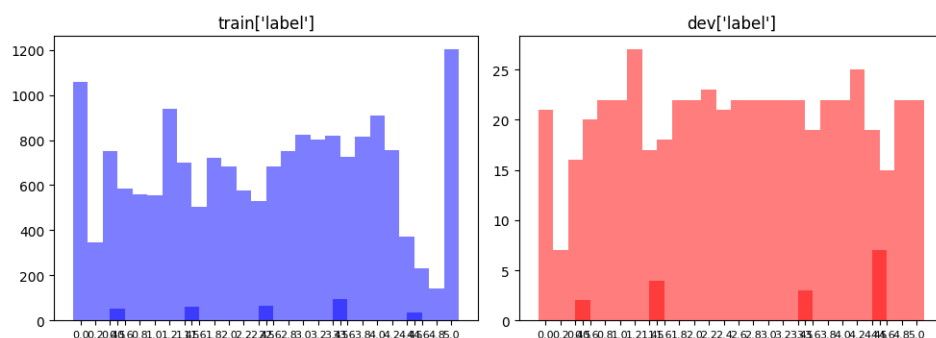


bert 증강기법으로 증강하였으나 여전히 label 0 데이터의 비율이 높았다.

따라서 label을 uniform, random 분포로 만들어주기 위해, label 0의 데이터를 잘라내서 label 5의 데이터로 만들어주는 작업을 진행했다.

이 때, copied sentence와 hanspell을 이용하여 label 0의 데이터 중 문장2 데이터를 hanspell 적용하여 label 0 개수와 비슷해질 때까지 문장2로 copy하여 label 5로 만들어주었다. 잘라내는 label 0 데이터의 비율을 조정하며 최적의 값을 찾아보았다.

후에 dev 분포와 비슷하게 같은 방법으로 증강이 필요한 label을 2배 증강하였다.



- Kolmogorov-Smirnov test : 귀무가설을 기각하지 못하므로, 분포가 비슷하다.

KS Statistic	0.046486
P-value	0.192672

- Swap sentence

두 문장의 순서를 바꾸는 방법이다. 문장 순서가 바뀌어도 유사도(label)는 동일함이 객관적으로 보장되며, 또 Bert 모델계열 사용 시에 Segmentation Embedding 값이 다르므로 유의미한 데이터 증강이 될 것이라 분석하였다. 그러나 성능이 오히려 떨어지는 결과가 나와 채택하지 않았다.



- K-fold


데이터 개수가 적을 때 일반화 성능의 향상을 위해 유용하게 쓸 수 있는 방법이다. 회귀문제는 stratified를 K-fold 쓸 수 없어 balanced하게 데이터 전처리 후 K-fold를 적용해보았으나, 데이터가 많이 증강된 후라 성능 향상에 크게 의미가 없었다.

4-3. 결과

Public Score 0.9285, Private Score 0.9349

학습 데이터와 검증 데이터의 분포의 차이를 줄이는 것, 모델의 Inference 결과를 분석하여 편향된 Inference 경향을 줄이는 것에 집중하여 대회 진행. 모델 측면에서 앙상블을 통해 일반화 성능을 높이고 편향된 Inference 경향을 줄이는 데 성공했다. 하지만 더 낮은 점수였던 모델의 Private Score가 더 높았다. 데이터 측면에서는 다양한 데이터 전처리, 증강 방법에 따른 모델 결과에 대한 분석이 부족했다.

53		0.9285 → 0.9349	상세 보기	2023-12-21 18:20			
52		0.9280 → 0.9353	상세 보기	2023-12-21 17:48			

8 (1 ▲)	NLP_09조	    	0.9349
------------	---------	--	--------

5. 자체 평가 의견

잘했던 점

- 핵심 키워드 분리 및 작업 분담
- 각자 맡은 부분들에 대하여 **task**별 정리

시도는 했으나 잘 안된 것들

- **General and Adaptive Robust Loss Function** 을 통해 모델과 데이터에 맞게 손실함수를 적응시켰지만 오히려 성능이 감소.
- 시간 부족으로 손실함수 실험을 하나의 모델에 대해서만 수행하여 다른 모델에 일반화할 수 있는 결과를 얻지 못함.
- 여러 가지 모델 학습(**S-bert**, **KcELECTRA**, **sroberta**)

아쉬운 점

- 실험 시에 검증 계획을 함께 수립하지 않아 제출 횟수 부족으로 검증하지 못한 실험들 존재.
- 앙상블에 대한 이해도가 높지 않아 최종 제출한 앙상블 모델보다 제출하지 않은 앙상블 모델이 더 좋은 결과를 냈다.

배운 점

- **huggingface** 기반 모델 로드 및 학습
- **wandb sweep**으로 하이퍼파라미터 튜닝하는 방법
- **pytorch lightning**을 이용한 모델 구현 및 구조 이해

6. 개인 회고

김인수 T6035 캠퍼

- 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

Pytorch Lightning 구조에 대해 공부하였고, 하이퍼파라미터 튜닝과 모니터링을 위한 Sweep, Wandb 적용하였다. 또한 학습데이터의 데이터 불균형을 해결하기 위한 방법을 조사, 적용하였다. 모델 앙상블에 대해 조사, 적용하고 적절한 모델을 선택하는 기준에 대해 고민하였다.

- 팀내 역할

Wandb, Sweep 등을 적용한 베이스라인 코드를 작성하여 Github에 공유하였다. 또한 모델의 하이퍼파라미터 튜닝과 앙상블 모델을 적용하였다. 학습데이터의 라벨 불균형을 해소하기 위해 Bert 모델을 통해 데이터 코드를 공유하였다.

- 마주한 한계와 아쉬운 점

대회 후반부에는 모델 성능 향상에 상한선이 정해져있듯 어떤 방법을 시도해봐도 성능이 오르지 않는 한계를 마주했다. 이때 하이퍼파라미터 튜닝에 몰두한 나머지 근본적인 학습 데이터에 대한 고민이 부족했다. 딥러닝 테크닉에 매몰되어 가장 중요한 점을 놓쳐 추가적인 성능 향상이 이루어지지 않는 이유를 파악하지 못했던 것 같다. 그렇기 때문에 프로젝트 정의부터 데이터 탐색 단계를 중요하게 생각하고 탐색하는 습관을 들여야 할 것 같다.

- 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

데이터 EDA에 시간을 투자하여 자세하게 살펴볼 것이다. 데이터의 특성을 파악하고 빠르게 베이스라인의 Inference 결과를 분석하여 대회 진행 방향성을 설정하여 다시 데이터에 대해 고민 하지 않도록 할 것이다. 또한 실험 전 실험 이유와 가설을 명확하게 설정한 후 실험을 수행하고, 실험들의 버전 관리에 신경써 정리할 수 있도록 할 것이다. 그리고 행동에 대한 근거를 찾기 위해 현재 상황에 대해 분석하는 태도를 가질 것이다.

- 어떤 방식으로 모델을 개선했는가?

모델 측면에서는 하이퍼파라미터 튜닝을 통해 단일 모델들의 성능을 높였으며, 앙상블 모델을 적용하여 일반화 성능을 향상시켰다. 또한 학습데이터를 다양화(증강 훈련 데이터, 증강 X 훈련 데이터)한 모델을 앙상블하여 성능을 높였다. 데이터 측면에서는 BERT를 통한 증강 기법을 통해 생성한 학습데이터로 학습하여 유의미한 모델의 성능 향상을 가져왔다.

문지원 T6052 캠퍼

- 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

Pytorch lightning 베이스라인 코드를 공부했으며, 베이스라인 코드에 **WandB Sweep**을 사용하여 하이퍼파라미터 튜닝하는 법을 배웠다. 또한 한국어 기반 모델을 조사하고 **WandB Sweep**을 활용해 성능을 비교했으며, 모델 앙상블의 구현 방식을 이해했다.

- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

공부한 개념을 코드로 구현하는 데 시간이 걸려서 직접 작성하지 못하고 대부분 팀원들이 공유한 코드를 기반으로 공부하고 수정해서 사용했다. 또한 빠르게 원하는 성능을 내지 못하거나 이해도가 부족해 구현이 어려운 실험들은 프로젝트 일정상 일찍 마감하면서도 더 깊이 있게 고민해서 도전해봤으면 하는 아쉬움이 남았다.

- 어떠한 실패를 경험했는가? 실패의 과정에서 어떠한 교훈을 얻었는가?

같은 의미의 문장 표기가 다른 경우 유사도 예측 성능이 떨어지는 것을 보고 데이터 전처리를 통한 성능 향상을 시도해보고자 했다. **Okt** 형태소 분석기의 어근 추출 기능을 이용하면 사용되는 표현을 표준화할 수 있을 것이라고 생각하고 실험해본 결과 오히려 적용하기 전에 비해 확연히 낮은 성능을 보였다. 데이터의 변형이 너무 커서 유의미한 부분까지 소실되었을 거라 해석했고, **Mecab** 형태소 분석기를 활용하여 다시 한번 실험했으나 일부 모델에서만 성능 향상이 있었고, 전체적으로 일관된 결과는 얻지 못했다. 실험 전 가설을 확실히 세우고 근거를 명확히 하는 고민이 부족했으며 데이터와 사용하는 모델에 대해 충분히 이해하지 않은 채 진행하다보니 이러한 결과가 나왔다고 생각된다.

- 어떤 방식으로 모델을 개선했는가?

팀 회의 시 다양한 데이터로 학습한 모델의 앙상블을 시도해보자는 의견이 나왔다. 이에 더해 앙상블할 모델의 증가를 통해서도 성능을 향상할 수 있을 것이라고 생각하고 증강 데이터를 사용하여 당시 가장 높은 성능을 냈던 모델 두 가지와, 그 중 한 모델로 기본 데이터를 학습한 것, 아직 활용하지 않았던 새로운 모델로 기본 데이터를 학습한 것, 총 4개의 모델을 앙상블했다. 그 결과 피어슨 상관 계수를 **0.9265**에서 **0.9280**으로 향상시키며 모델의 추가적인 개선 가능성을 확인했다.

- 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

프로젝트 기획과 데이터 분석에 더 시간을 들이며 구체적인 실험 가설과 목표를 설정하고 진행해야겠다고 느꼈다. 이를 통해 더 효과적으로 실험을 진행하고 필요 이상으로 소모되는 시간을 줄이고자 한다. 또한 프로젝트와 모델의 체계적인 버전 관리가 필요함을 느꼈고, 이번 프로젝트에서는 시간상 제대로 활용하지 못한 깃헙 기반 협업을 시도해보고 싶다.

양서현 T6099 캠퍼

- 학습목표를 달성하기 위해 무엇을 어떻게 했는가?

대회 참여가 처음이었기 때문에, 모델의 좋은 성능도 중요하지만 프로젝트의 전반적인 흐름파악과 프로젝트 이전에 학습했던 내용을 활용해보는 것, **learning by doing**을 목표로 잡았다. 특히 EDA(Exploratory Data Analysis)와 데이터 불균형을 해결하기 위한 **Data Augmentation** 작업에 집중적으로 참여하였다.

● 참여 Task

- EDA
- Data Augmentation - copied sentence, label smoothing, swap sentence
- Data Cleaning - 특수문자제거(regex), 맞춤법 교정(PyHanspell)
- Validation - K-fold

- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

swap sentence 같은 기법을 사용하여 데이터 증강을 시도했을 때, 성능이 당연히 개선되어야 한다고 생각했던 부분에서 오히려 성능이 하락하는 경우가 많았다.

- 어떤 방식으로 모델을 개선했는가?

- 데이터 측면

data preprocessing 과정을 통해 성능을 높일 수 있는 최적의 데이터셋을 선별하였다. **label**의 **uniform, random** 분포를 위해, **label 0**의 데이터를 잘라내서 **label 5**의 데이터로 만들어주는 작업을 진행했다. 이 때, **copied sentence**와 **PyHanspell**, 특수문자 제거를 적용하여 **label 0**의 데이터 중 문장2 데이터를 **label 0** 개수와 비슷해질 때까지 문장2로 **copy**하여 **label 5**로 만들어주었다. 잘라내는 **label 0** 데이터의 비율을 조정하며 최적의 값을 찾아보았다. 최종적으로는 피어슨 상관 계수를 향상시키며 모델의 성능 향상에 기여하였다.

- 모델 측면

다양한 단일 모델을 적용하며 성능을 파악하였다. 최적의 앙상블 모델을 찾아가며 일반화 성능을 향상시켰다.

- 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

단순히 데이터 증강이나 모델의 성능을 높이려 한 시도는 많았으나, 데이터셋의 분포나 변수 파악 외에 데이터 자체를 들여다보거나 하이퍼 파라미터를 튜닝하는 데에 시간을 많이 투자하지 못했다. 이번 프로젝트의 경험을 교훈 삼아 다음 프로젝트에서는 이와 같은 부분을 진행 해보면서 더 나은 프로젝트로 발전시키고 싶다.

오주영 T6103 캠퍼

- 학습목표 달성을 위해 수행한 것:

- 베이스라인 코드를 따라서 다시 짜보고 **Early stopper, checkpoint** 등을 적용하며 모델의 설계, **Pytorch Lightning(PL)**의 기능과 적용에 대한 이해를 높였다.
- **PL**에서 지원하는 손실함수의 종류와 목적을 분석하고 **STS** 학습에 적합해 보이는 손실함수들을 선정했다. 더해 적응형 손실함수와 다중 손실함수에 관한 논문을 읽고 실험에 적용했다.

- 모델 개선 방법과 결과:

- 하나의 모델에 대해 다양한 손실함수를 적용함으로써 손실함수의 종류가 모델의 학습과 일반화 능력에 미치는 영향에 대해 탐구했다. 실험한 손실함수들은 초기 설정인 **L1Loss** 와 대등하거나 높은 성능을 보였다.
- **MSE** 손실함수의 경우 모델에 무관하게 일반적인 성능향상을 보였으며 손실함수에 적절한 **noise**를 추가했을 때 모델의 일반화 성능이 높아질 수 있다는 가능성을 확인했다.

- 마주한 한계와 아쉬웠던 점:

- 적절한 모델 선정과 **hyper parameter** 값 선정을 선행해야 했다. 팀내 합의가 이루어지기 전에 실험을 시작해 최종 결과물에 반영이 어려웠다.
- 실험 계획과 검증 계획을 동시에 수립하지 않아 수행한 실험을 모두 검증하지는 못했다.
- 하드 스킬의 부족으로 다양한 아이디어의 적용에 어려움이 있었다. (사고의 속도가 구현의 속도에 의해 제한되었다.)
- 유기적인 협업을 수행하지 못했다.

- 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것:

- 깃허브를 이용한 이슈기반의 협업
- 데이터 자체에 대해 깊게 탐구하기
- 코드의 재사용성 높이기
- 목적에 맞는 임베딩 공간 튜닝
- 학습과정의 노이즈가 모델 일반화 성능에 미치는 영향에 대해 일반화할 수 있는 결론 도출하기.

- 총평:

- 모델학습을 경험하며 학습의 흐름을 직접 느낄 수 있었다.
- 가설의 설정과 실험 설계, 검증의 단계를 거치며 다소 지식간의 연결이 긴밀하지 않았던 부분이나 미처 고려하지 못했던 부분들을 발견할 수 있어 유익했다.
- '이랬으면 더 좋았겠다' 하는 지점들이 많다. 이를 다음 프로젝트에 잘 녹여낼 수 있도록 잘 가져가자.

손윤환 T6084 캠퍼

- 나는 내 학습목표를 달성하기 위해 무엇을 어떻게 하였는가?

프로젝트에 본격적으로 임하기에 앞서, 프로젝트 주제와 주어진 베이스라인 코드의 이해에 중점을 두고 시작하였다.

그 이후로 데이터 분포와 특성에 대하여 이해하려고 노력하였고, 한국어 기반 사전학습 모델들의 성능이 천차만별인 것을 발견하고 여러 모델들을 **huggingface**, **github**에서 탐색하였다.

마지막으로는 주어진 데이터에 대한 결과에서 이상치에 대하여 분석하고, 데이터 증강을 통하여 모델을 개선하였다.

- 전과 비교해서, 내가 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

과거에 **AI**를 배우고, 실습을 진행하였을 때에는 하이퍼파라미터 튜닝, 더 나은 모델 탐색, 데이터 증강 및 상세분석 등을 진행하지 않고 데이터에만 의존하는 경향이 있었다.

하지만 이번 기회를 통하여 위에서 말한 여러 작업들을 직접 수행해보고, **W&B**를 통하여 실제 학습진행에 대한 과정을 눈으로도 확인해보며 **AI** 모델 학습 전반에 대해 한층 성장하는 계기가 되었다.

- 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

주로 데이터와 관련하여 분석하는 과정에서 시각화 방법이나, 분석 과정에 대한 퍼포먼스가 아쉬웠다. 데이터 분석에 관한 반복적인 연습과 경험을 통해 개선해나가야 할 큰 숙제이다.

이번 프로젝트를 진행하면서, 주어진 작업에 대해서 참여는 하였지만, 자기만족도가 높지 않은 것을 보아하니, 더 열심히 참여했으면 더 좋은 결과가 있었을 것 같다.

- 한계/교훈을 바탕으로 다음 프로젝트에서 스스로 새롭게 시도해볼 것은 무엇일까?

이번 프로젝트 겸 대회를 통하여 많은 것을 경험해보는 시간을 가졌다. 나에 대해서 장/단점이 무엇이고, 어떻게 개선을 해나가면 좋을지 탐색하는 귀중한 시간이었다.

여러 프로젝트를 거치며 실력을 계속 갈고닦는 **AI** 엔지니어가 될 수 있도록 초심을 잃지 않는 자세로 배우며, 성장해나가야 한다는 깨달음을 얻었다.

AI에 있어서는 데이터는 가장 중요한 부분이라는 생각이 들었고, 다음 기회가 있다면 더 많은 방법과 수단을 동반하여 데이터와 관련된 작업들을 많이 진행할 것 같다.