

Semantic Text Similarity Wrap-up Report

NLP-01조(1조라도 안보이면)

1 프로젝트 개요

1. 프로젝트 주제 및 목적

- STS(Semantic Text Similarity)란 두 텍스트가 얼마나 유사한지 판단하는 NLP Task로, 일반적으로 두 개의 문장을 입력하고 이러한 문장 쌍이 얼마나 의미적으로 서로 얼마나 유사한지를 판단하는 과제이다.
- 본 프로젝트는 주어진 데이터셋을 바탕으로 0과 5사이의 유사도 점수를 예측하는 모델을 만드는 것에 목적을 둔다.

2. 프로젝트 환경

컴퓨팅 환경	5인 1팀, 인당 V100 서버를 VSCode와 SSH로 연결하여 사용
협업 환경	Notion, GitHub, WandB, Google Drive
의사 소통	Slack, Zoom, 카카오톡

3. 프로젝트 구조

a. 데이터 총 개수

- Data - 총 10,974 문장 쌍
 - Train 9,324
 - Validation 550
 - Test 1,100

b. 데이터셋 구조

Column	설명
id	문장 고유 ID. 데이터의 이름, 버전, train/dev/test
source	문장의 출처 - petition(국민청원), NSMC(네이버 영화), slack(업스테이지)
sentence_1	문장 쌍의 첫번째 문장
sentence_2	문장 쌍의 두번째 문장
label	문장 쌍에 대한 유사도 (0~5, 소수점 첫번째 자리까지 표시)
binary-label	label이 2.5 이하인 경우는 0, 나머지는 1

c. Label 점수 기준

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

d. 평가 지표: Pearson Correlation (피어슨 상관계수)

e. 대회 중 리더보드 평가 기준: Public Score (Test의 50%를 바탕으로 평가)

f. 최종 평가 기준: Private Score (Test의 100%를 바탕으로 평가)

g. 하루 제출 횟수: 10회

2 프로젝트 팀 구성 및 역할

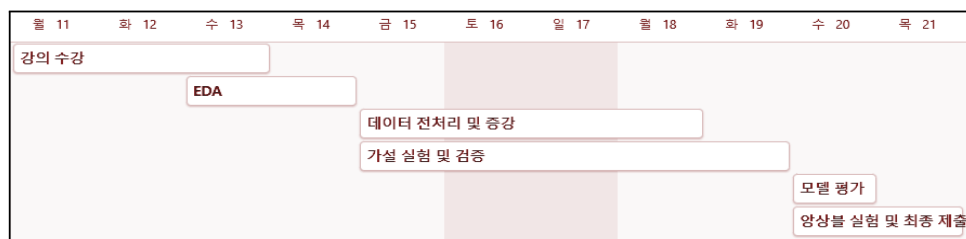
1. 역할

이름	담당 업무
전현욱	팀 리더, ensemble 구현, 단일 모델 학습
곽수연	Weighted Sampler 구현, 단일 모델 학습
김가영	loss function 실험, 단일 모델 학습
김신우	복합 모델 실험, K-Fold 구현, 단일 모델 학습
안윤주	데이터 전처리 및 증강, 단일 모델 학습

3 프로젝트 수행 절차 및 방법

1. 프로젝트 기간

- 2023-12-11 10:00 ~ 2023-12-21 19:00



2. 활용된 기술 및 라이브러리

- 개발 언어: Python
- 데이터 전처리 및 증강: Pandas, Numpy, GoogleTrans, Request, PyKoSpacing
- 모델링: PyTorch, Scikit-Learn, HuggingFace
- 성능 분석: WandB

3. 프로젝트 세부 수행 절차

- 1) 2023-12-13 (수) 까지 대회 기초 강의 전부 수강 완료
- 2) 서버 수령 후 GitHub 및 작업 환경 설정
- 3) 대회 개요 및 데이터 성질, 베이스라인 코드 분석
- 4) EDA 진행 후, 분석 내용을 바탕으로 가설 설정
- 5) 모델 선정 및 역할 분담 (데이터 전처리 및 증강 1人, 가설 시험 4人)
- 6) 데이터 전처리 및 증강
- 7) 가설 실험 및 검증 (5번과 동시 진행)
- 8) 검증 결과를 바탕으로 최종 방법 선택 후 모델 학습 및 평가
- 9) 평가한 모델 성능 비교 후 Ensemble 진행
- 10) 최종 결과물 제출

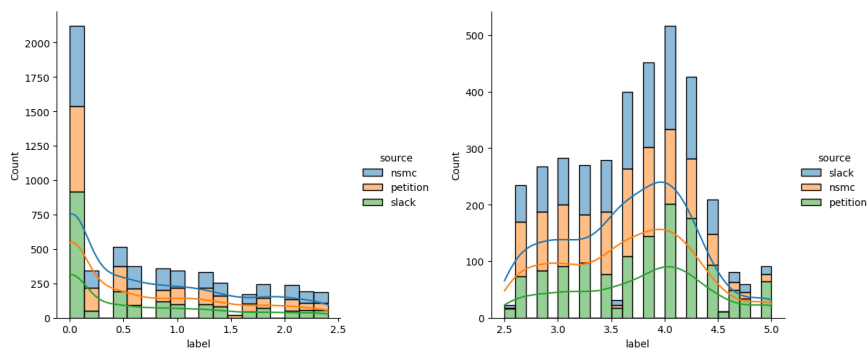
4 프로젝트 수행 결과

1. 작업 환경 설정

- GitHub : branch 별로 버전 관리
- V100 Server : 코드 작성 및 모델 학습
- WandB : 팀 프로젝트 공간에서 모델 성능 분석

2. EDA

- Label 분포 확인 : 0점대의 label이 다른 label 보다 개수가 2배 이상 많았다.



▲왼쪽은 binary-label이 0인 데이터 분포, 오른쪽은 binary-label이 1인 데이터 분포

- 문장 쌍 텍스트 데이터 확인 : 어떤 label에 어떤 문장 쌍이 있는지 확인

3. 가설 수립 및 검증

- 데이터 증강을 하면, 성능이 향상될 것임.
- [UNK] 토큰을 톡아본 결과, 오타가 다수 존재함. 따라서, 오타 교정을 시도하면 토큰나이징에 도움이 되어, [UNK] 토큰이 줄어 들기 때문에 성능이 개선될 것임.
- 종속변수(target)를 구간화하면, 예측 범위가 줄어들어 성능이 개선될 것임.
- source 정보를 토큰화 하여 넣으면, 예측의 성능을 높이는 데 효과적일 것임.
- binary-label의 값을 활용하여, 0과 1의 모델을 각각 학습시킨 뒤 예측한 결과를 ensemble 하면 성능 개선에 도움을 줄 것임.
- uniform한 분포를 갖추면 모델이 robust 해질 것임.

4. 데이터 전처리

- 한글, 알파벳, 숫자 제외 특수기호 제거
- 알파벳 대문자를 소문자로 치환
- 띄어쓰기 교정
- 종속변수 카테고리화

5. 데이터 증강

- 딥러닝 학습을 보다 효과적으로 하기 위해 데이터 증강이 필요하다고 판단. 따라서, 아래와 같은 방법으로 시도.

1) 문장 도치

- 문장을 띄어쓰기 기준으로 반으로 나눠, 앞뒤 순서를 도치

예시	sentence1	sentence2	Label
원본 문장	미세먼지 해결이 가장 시급한 문제입니다	가장 시급한 것이 신생아실 관리입니다	0.4
변경 문장	가장 시급한 문제입니다 미세먼지 해결이	것이 신생아실 관리입니다 가장 시급한	0.4

2) 맞춤법 교정

- 부산대 맞춤법 검사기를 활용해 맞춤법 교정

예시	sentence1	sentence2	Label
원본 문장	아무리그래도 걸어간다는설정은쫘	허나 알리슨 로먼이 벗은건 충격	0
변경 문장	아무리 그래도 걸어간다는 설정은 쫘	하지만 알리슨 로먼이 벗은 건 충격	0

3) 역번역(back translation)

- GoogleTrans를 활용해 한국어를 영어로 번역 후, 번역된 문장을 다시 한국어로 번역

예시	sentence1	sentence2	Label
원본 문장	보험이 재산압류를 해서는 안 될 일입니다.	보험은 재산을 몰수해서는 안됩니다.	3.4
변경 문장	보험을 압수해서는 안됩니다.	보험을 압수해서는 안됩니다.	3.4

4) under sampling + swap sentence + copied sentence + uniform

- 0의 데이터 일부를 drop한 후, drop한 데이터 일부의 문장 쌍을 같게 하여 5의 데이터로 사용
- 각 label의 평균 개수에 맞춰서 위의 증강 기법 + sentence swapping을 적용해서 최대한 uniform한 분포를 가지도록 작업

5) distribution + random noise

- 데이터수가 상대적으로 적은 0.5~3.5 데이터에 대해 오타를 섞은 데이터 삽입
- 데이터수가 상대적으로 적은 0.5~3.5 데이터에 대해 공백을 섞은 데이터 삽입

6. 모델 선정

- Baseline 기준으로 성능이 높은 모델 위주로 선정
 - klue/roberta-small
 - klue/roberta-large
 - rurupang/roberta-base-finetuned-sts
 - monologg/koelectra-base-v3-discriminator
 - BM-K/KoDiffCSE-RoBERTa
 - snunlp/KR-ELECTRA-discriminator

7. 모델 학습 및 성능

- WeightedRandomSampler : label이 batch마다 비율이 같게 나오도록 설정
- K-Fold : 5개의 fold로 split
- Attention Mask : padding을 했기 때문에 사용
- Learning Rate Scheduler : Cosine Annealing

1 assemble[assemble.std(axis=1) > 0.6]

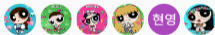
	koelec	koelec-a	elec-fold
46	2.854490	1.530521	3.046300
195	1.943579	0.906262	2.031131
358	3.465977	2.251090	3.549714
383	3.068318	1.472543	2.626327
401	2.376429	1.184119	2.176941
436	2.874614	1.433549	2.601663
516	2.317434	1.094975	2.084260
658	1.797669	0.829282	1.985682
682	2.602845	1.192044	2.588484
704	0.311952	1.630754	0.871383
786	1.148815	2.306225	2.359981
788	1.945525	0.807594	1.804124
994	2.443039	0.660699	1.959610
1022	1.544263	0.608494	1.731864
1048	2.276085	0.958804	2.518114

8. Ensemble 실험

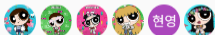
- 유의미한 ensemble을 만들기 위해 모델의 예측값의 분산을 행 기준으로 분석
- 모델의 예측 경향성을 파악해서 경향성이 비슷한 모델끼리 ensemble 진행

9. 최종 결과

[Public Score] : 0.9218

11 (-)	NLP_01조		0.9218	40	1d
-----------	---------	---	--------	----	----

[Private Score] : 0.9311

10 (1 ▲)	NLP_01조		0.9311	40	1d
-------------	---------	---	--------	----	----

[최종 채택한 방법론 및 모델]

- 데이터 전처리 및 증강
 - 사용한 최종 데이터의 개수 : 약 3만 3천개
 - 추후 가공 데이터 : 약 2만 8천개
- 모델
 - 최종 ensemble 모델
 - 1) snunlp/KR-ELECTRA-discriminator
 - 2) klue/roberta-large
 - 3) BM-K/KoDiffCSE-RoBERTa
 - 4) monologg/koelectra-base-v3-discriminator
- 모델 학습
 - K-Fold(5 split) + Weighted Random Sampler
 - Learning Rate Scheduler + Attention Mask