


Text Classification project



TEAM 7조 LSTedM
김주한, 윤현준, 류다빈, 박지원, 이진국



목차

1. 프로젝트 개요
 2. 프로젝트 팀 구성 및 역할
 3. 프로젝트 진행 프로세스
 4. 프로젝트 결과
 5. 자체 평가 및 보완
- 

프로젝트 개요

주어진 문장 분류기의 정확도를 올리는 프로젝트이다.

영문으로 이루어진 음식점 리뷰 데이터를
긍정과 부정으로 카테고리화 하여 분류한다.

Baseline Code를 분석하고 성능을 개선시킨다.

활용 라이브러리

표준 라이브러리

- Os
- Pdb
- Argparse
- Dataclasses
- Typing
- Collections

외부 라이브러리

- Wandb
- Torch
- Numpy
- Tqdm
- transformers



프로젝트 팀 구성 및 역할



김주한 | 팀장

기존 Baseline Code 분석

GitHub repository 생성 및 관리

Kaggle 팀 관리



류다빈 | 팀원

기존 Baseline Code 분석

자료 수집 및 공유



윤현준 | 팀원

기존 Baseline Code 분석

문장 분류기 정확도 개선



박지원 | 팀원

기존 Baseline Code 분석

자료 수집 및 공유

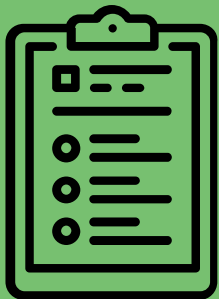


이진국 | 팀원

기존 Baseline Code 분석

발표자료 작성





프로젝트 진행 프로세스

기간	활동
10/ 25(월)	주어진 Baseline Code 분석 및 문제점 파악
10/ 26(화)	주어진 Baseline Code 분석 및 문제점 파악 Github Repository 생성
10/ 27(수)	기존 index sorting Code 수정 Kaggle 팀 구성
10/ 28(목)	Batch Size, Hyper Parameter 변경
10/ 29(금)	최종점검 프로젝트 결과물 제작

프로젝트 결과

성능 개선 방법

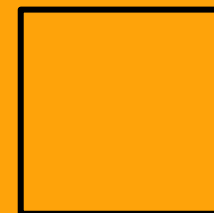


데이터 모델링

Pre trained model 변경
데이터 양을 늘림

Hyper Parameter 변경

Batchsize, epochs, learning rate,
random seed, optimizer, weight
initialization, Early stop strategy,
Regularization, Dropout 등을 변경
및 조정



프로젝트 결과 이슈



초기 베이스라인 코드를 실행하여 성능을 테스트했을 때 제대로 된 성능이 나오지 않음

원인

index를 sorting 해주어서
csv 파일에 inference할 경우
test data의 index가 바뀌어서
성능이 떨어지는 현상 발생

해결 방안

collate_fn_sentiment_test()
function에서 기존 샘플의
index를 sorting하는 것이 아닌
단순 ndarray 형태로
넘겨주어 해결

결과

0.4985 -> 0.98로
성능개선

프로젝트 결과 이슈

구분	Baseline code 성능 테스트	Batchsize를 줄이고 epoch를 늘릴 경우 시간이 오래 걸리고 성능이 조금 개선되나 큰 변화 없음	Train 하는 batchsize를 늘리고 나머지 조건은 2차 시도와 동일할 경우 성능이 좋아짐	Epoch를 4로 늘릴 경우 성능이 많이 좋아짐
Batchsize	64	32	128	128
Randomseed	42	40	40	42
Learning rate	5e-5	5e-5	5e-5	5e-5
epoch	1	2	2	4
score	0.98	0.981	0.984	0.988



자체 평가 및 보완

팀 의견

Baseline code없이 진행해 보고싶다.

전체적인 흐름을 볼 수 있어서 좋았다.

팀원 간의 교류가 더 활발했으면 좋겠다.

자연어 처리에 관한 공부 부족함을 느꼈다.

