Week1-4 과제

## 한국 스트리밍 서비스 (왓\*, 쿠\*플레이, 티\*)에서 시청자가 영화를 보고 남긴 리뷰를 긍정과 부정으로 나누어 볼 수 있는 대시보드를 만들려고 한다. **리뷰 긍부정 판별 모델**을 만들려고 할 때, NLP 리서처/엔지니어로서 어떤 의사 결정을 할 것인지 각 단계에 맞춰 작성해보자. (단, 수집된 리뷰 데이터의 개수가 1,000개 미만이라고 가정하자.)

대시 보드 예시.

| 긍정 | 부정 |
| --- | --- |
| ID:  REVIEW: | ID:  REVIEW: |
| ID: REVIEW: | ID: REVIEW: |

### 문제 정의

풀고자 하는 문제를 정의하세요. 또한 데이터 생성 시 고려해야할 사항이 있다면 무엇인지 설명하세요. (예, 만약 긍정 리뷰가 부정 리뷰보다 많은 경우 어떻게 해야 할까?, 길이가 정말 긴 리뷰는 어떻게 전처리 해야 할까?)

—-

* **문제정의**  
  1. 리뷰 데이터를 기반으로 긍정과 부정을 판별  
  2. 긍정과 부정에 해당되는 리뷰를 나눠 볼 수 있는 대시보드 생성
* **데이터 생성 시 고려사항**

0. 수집해야되는 데이터

- 리뷰, 평점, ID(대쉬보드 출력을 위함)

1. 리뷰의 label을 직접 판단해야하는 경우 긍정과 부정을 판단하는 기준이 필요함.

- ex) 평점 9~10점: 긍정 / 평점 1~4점: 부정

2. imbalanced data

- undersampling: 적은 label의 데이터에 맞춰서 많은 label의 데이터를 삭제  
- oversampling: 많은 label의 데이터에 맞춰 적은 label의 데이터를 증가(Generation, 랜덤으로 데이터를 복사, SMOTE(Synthetic Minority Over-sampling Technique)을 통한 생성 등)

3. 길이가 긴 데이터

- 텍스트를 길이에 맞게 잘라낸다.  
(How to Fine-Tune BERT for Text Classification? 논문에 따르면 문장의 중간을 잘라내는게 효과가 가장 좋다고 연구되었다.)  
- 긴 텍스트를 문장 단위로 나누고 각각 하위 문장 단위로 새롭게 label을 적용한다.  
- 정해진 길이에 맞춰 모델(embedding)을 늘리고 각각의 모델의 결과(vector)를 합쳐서 새로운 모델(classifier)을 학습

4. 전처리

- 감정 분석에 영향이 없는 또는 악영향을 끼치는 데이터를 처리한다.  
 (ex: 열맞춤을 위한 공백 삭제, 한자->한글 치환, 이메일/전화번호 삭제 등)

5. 너무 적은 양의 데이터

- Transfer Learning을 통해 학습을 해야하므로 유사한 데이터를 Pre-trained한 모델을 탐색한다.

### 오픈 데이터 셋 및 벤치 마크 조사

리뷰 긍부정 판별 모델에 사용할 수 있는 한국어 데이터 셋이 무엇이 있는지 찾아보고, 데이터 셋에 대한 설명과 링크를 정리하세요. 추가적으로 영어 데이터셋도 있다면 정리하세요.

—-

**한글**

네이버 영화 리뷰에 대한 긍정/부정 데이터(NSMC)

설명:

* 영화 리뷰에 대한 text와 해당 리뷰의 긍정(1), 부정(0)을 표시한 label로 구성  
  리뷰의 평점을 기준으로 label 측정: 긍정(9~10점), 부정(1~4점)
* 200K개 리뷰로 구성된 데이터 (Train: 150K, Test: 50K)
* 모든 리뷰는 140자 미만

링크: <https://github.com/e9t/nsmc>

**English**

SST-2(Stanford Sentiment Treebank Binyary)

설명:

* GLUE에 포함된 데이터
* Rotten Tomatoes의 영화 코멘트 데이터로 긍정(1), 부정(0)을 표시
* Train 67K, Dev 872, Test 1.8K로 구성된 데이터

링크: <https://deepai.org/dataset/stanford-sentiment-treebank>

### 모델 조사

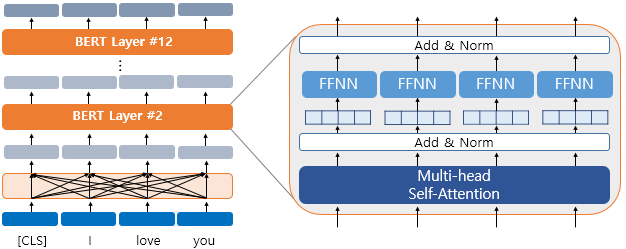
Paperswithcode(https://paperswithcode.com/)에서 리뷰 긍부정 판별 모델로 사용할 수 있는 SOTA 모델을 찾아보고 SOTA 모델의 구조에 대해 간략하게 설명하세요. (모델 논문을 자세히 읽지 않아도 괜찮습니다. 키워드 중심으로 설명해 주세요.)

—-

**한글(NSMC)**

KcBERT(Acc. 90.7%)

* BERT 모델 사용
* 댓글의 특수성(신조어, 줄임 표현)을 고려해 네이버 댓글을 pre-training 데이터로 사용함
* BERT 구조



* 현재 업데이트 모델 KcELECTRA가 공개(Acc. 91.7%)

**영어(SST-2)**

SMART-RoBERTa Large(Acc. 97.5%)

* RoBERTa 모델에 Smoothness-Inducing Adversarial Regularization 이론을 적용한 방식
* RoBERTa  
  BERT 모델과 동일  
  Masking 단계에서 dynamic masking을 적용  
  (학습할 때 마다 masking을 새로 진행하는 방식)  
  NSP(Next Sentence Prediction)을 없애고 512 길이에 맞춰 하나의 문장으로 입력
* Smoothness-Inducing Adversarial Regularization  
  fine-tuning 단계를 효율적으로 하기 위한 정규화 이론

### 학습 방식

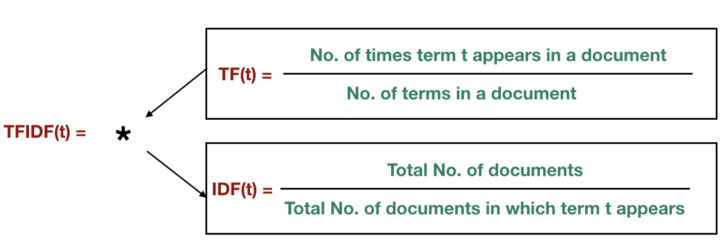
* 딥러닝 (Transfer Learning)  
  사전 학습된 모델을 활용하는 (transfer - learning)방식으로 학습하려고 합니다. 이 때 학습 과정을 간략하게 서술해주세요. (예. 데이터 전처리 → 사전 학습된 모델을 00에서 가져옴 → …)

1. Pre-trained (직접 생성할 경우)   
   유사한 데이터를 가져와 전처리를 한 후 모델을 학습한다.
2. 데이터 전처리 & labeling & 데이터 split  
   목적에 맞게 필요 없는 데이터를 지우거나 편집한다(ex: 한자, 과한 특수문자 등)  
   label 기준을 설정하고 데이터에 따라 기준에 맞는 label을 지정한다.   
   Train, Validation, Test 학습용 데이터를 나눈다.
3. Pre-trained 모델 import  
   1 또는 opensource로 공개된 pre-trained 모델을 불러온다.
4. fine-tuning 기능 구현  
   pre-trained된 모델을 3에서 전처리한 데이터로 fine-tuning할 수 있도록 기능을 구현한다.
5. 모델 최적화(parameter tuning)  
   batch size, epoch 등 파라미터를 튜닝하면서 가장 좋은 성능을 보여주는 파라미터를 찾는다.
6. API 또는 output 생성(ex: classifier, 대시보드)  
   목적(긍/부정 분류, 대시보드 생성)에 맞는 output을 만든다.

* (Optional, 점수에 반영 X) 전통적인 방식  
  Transfer Learning 이전에 사용했던 방식 중 TF-IDF를 이용한 방법이 있습니다. TF-IDF를 이용한다고 했을 때, 학습 과정을 간략하게 서술해주세요.

—-

1. 데이터 전처리 & labeling & 데이터 split  
   목적에 맞게 필요 없는 데이터를 지우거나 편집한다(ex: 한자, 과한 특수문자 등)  
   label 기준을 설정하고 데이터에 따라 기준에 맞는 label을 지정한다.   
   Train, Validation, Test 학습용 데이터를 나눈다.
2. 토큰화(Tokenization)  
   공백 문자 또는 문자 등 기준에 따라 문장을 토큰으로 나눔
3. TF-IDF 모델 생성

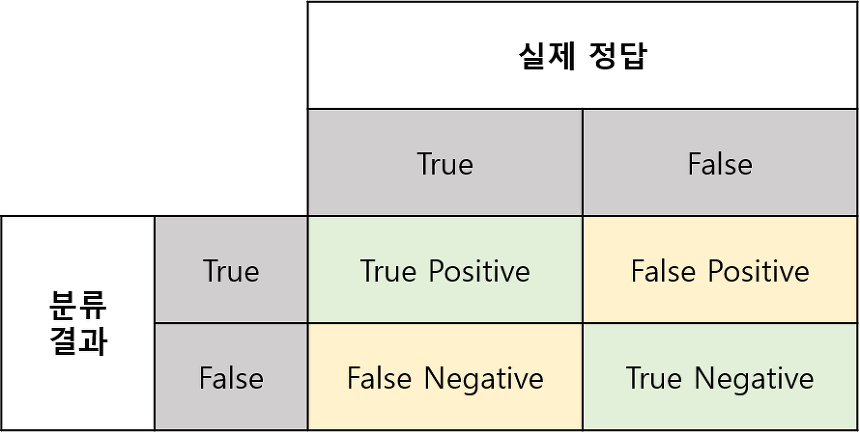


1. 토큰화된 데이터를 TF-IDF 모델을 사용해 벡터로 변환
2. 전통적인 머신러닝(DecisionTree, K-NN, LR, SVM 등)을 사용해 분류 모델을 생성 및 학습
3. API 또는 output 생성(ex: classifier, 대시보드)  
   목적(긍/부정 분류, 대시보드 생성)에 맞는 output을 만든다.

### 평가 방식

긍부정 예측 task에서 주로 사용하는 평가 지표를 최소 4개 조사하고 설명하세요.

—-



Accuracy: 모든 경우에서 옳게 예측한 것의 비율



Precision: 모델이 True라고 분류한 것 중에서 실제 True인 것의 비율



Recall: 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율



F1-score: Precision과 Recall의 조화평균

