# 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:<br>REVIEW: | ID:<br>REVIEW: |
| ID:<br>REVIEW: | ID:<br>REVIEW: |

1. 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?)

   —-

   **감성분석이란?**

   감성 분석이란, 주어진 글의 극성(긍정, 부정)을 분류하는 Natural Language Understanding의 하위 task입니다. 예를 들면 이 트윗이 긍정이다, 부정이다, 중립이다 처럼 분류하는 것입니다. 텍스트와 라벨이 주어지면 모델이 올바른 감정을 예측하도록 학습됩니다.

   **감성분석의 데이터 Imbalance**

감성 분석은 이진분류 문제이기 때문에, Imbalance 데이터로 학습시키면 Accuracy가 많이 틀릴 수 있습니다. 예를 들어 부정 데이터가 70%정도를 차지한다면 모든 테스트 데이터에 대해 부정으로만 예측해도 70%의 정확도를 가진 모델이 됩니다. 그러나 이 모델이 진정 70%의 예측 성능을 가진 모델이라고 할 수는 없습니다[1].

데이터 불균형 문제(Imbalance)를 해결하기 위한 대표적인 방법은 oversampling, undersampling, SMOTE, Semi-supervised learning 등이 있습니다[2].

- oversampling : 낮은 비율 클래스의 데이터 수를 늘림으로써 데이터 불균형을 해소하는 방법입니다. 데이터를 어떻게 생성하는지가 주된 문제가 됩니다[3].

- undersampling : 높은 비율을 차지하는 클래스의 데이터 수를 줄임으로써 데이터 불균형을 해소할 수 있습니다. 그러나 이 방법은 전체 데이터 수가 많이 줄어들기때문에 위 문제처럼 1000개 미만의 적은 데이터를 가진 경우 적합하지 않다고 생각됩니다[3].

- SMOTE : 낮은 비율 클래스 데이터들의 최근접 이웃을 이용하여 새로운 데이터를 생성합니다. 근접한 데이터들을 이용해서 현재 훈련 데이터셋에 포함되어있지 않은 데이터를 생성합니다[3].

그러나 감성분석은 텍스트 데이터이므로 일반적인 데이터를 oversampling할 때와는 다른 기법이 적용되어야합니다. [4]에 따르면 감성분석 라벨링을 하는 방법은 크게 수작업 라벨링, 평점을 따른 라벨링, 직접 감성사전을 구축하여 레이블된 학습데이터를 생성하는 방법 등이 존재합니다.

## 리뷰들 사이의 길이 차이가 큰 경우 전처리 방법

텍스트 데이터를 생성하거나 임베딩할 때 걸림돌이 되는 것은 긴 리뷰를 처리할 때입니다. 이런 task의 경우에는 십중팔구 RNN 기반의 모델을 사용하게 될 텐데, 리뷰가 길다면 먼 거리에 있는 단어들끼리의 관계성을 모델링하기가 어려워지기 때문입니다. 또한 padding을 할 때 의미없는 차원이 많아져서 예측을 잘 못할 수도 있습니다.

[5]에 따르면 short-text가 아닌 Document 수준의 감성 분석을 할 때는 RNN보다 CNN(Convolutional Neural Network) 기반의 모델이 더 좋은 성능을 보였다고 합니다. CNN을 이용하면 문서 전체의 극성을 더 잘 캐치해 낼 수 있다고 합니다. 이

사실에 기반하여, [5]는 CNN이 텍스트로부터 많은 feature를 추출할 수 있다는 사실을 이용해 CNN과 Bi-LSTM을 결합한 모델을 제시하기도 했습니다.

만약 리뷰가 짧은 글이 아닌 평론가가 쓴 글처럼 문서 단위라면 CNN을 결합한 모델을 사용해보는 것도 좋을 것 같습니다. 이 외에도 문장을 끊어서 새로운 데이터를 위에서 말한 라벨링 방법을 통해 새로운 데이터로 사용할 수도 있을 것 같습니다. 하지만 기본적으로는 Transformer 기반의 모델을 사용하면 길이차이가 아주 크지 않은 이상 왠만한 성능은 잘 보장하는 것으로 알고 있습니다.

## 데이터가 부족할 때는 전이학습 사용

데이터의 수가 부족하다면 기본적으로 가장 먼저 떠올릴 수 있는 것이 전이학습(Transfer Learning)이다. 전이학습이란 내가 풀고싶은 도메인(Target domain이라고 함)에서 데이터가 부족할 때, 다른 소스 도메인(Source domain)에서 학습된 모델을 가져와서 타겟도메인의 데이터로 fine-tuning하여 사용하는 방법이다. 정확히는 fine-tuning이라는 것이 전이학습의 한 가지 방법이다.

[1]https://towardsdatascience.com/approaches-to-sentimental-analysis-on-a-small-imbalanced-dataset-without-deep-learning-a314817e687

[2]https://hwi-doc.tistory.com/entry/%EC%96%B8%EB%8D%94-%EC%83%98%ED%94%8C%EB%A7%81Undersampling%EA%B3%BC-%EC%98%A4%EB%B2%84-%EC%83%98%ED%94%8C%EB%A7%81Oversampling

[3]https://hal.archives-ouvertes.fr/hal-01504684/document

[4]최민성, 온병원 (2019). Bi-LSTM 기반 감성분석을 위한 대용량 학습데이터 자동 생성 방안. 정보과학회논문지,46(8), 800-813

[5]Rhanoui, M., Mikram, M., Yousfi, S., & Barzali, S. (2019). A CNN-BiLSTM model for document-level sentiment analysis. *Machine Learning and Knowledge Extraction*, *1*(3), 832-847.

## 2. 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.

—-

### 1. 네이버에서 공개한 영화 리뷰 감성분석 데이터 nsmc(Naver sentiment movie corpus)

링크 : https://github.com/e9t/nsmc

설명 : 네이버 영화 페이지에서 수집된 한국어 영화 리뷰 데이터셋이다. 데이터의 변수는 id, document, label이 있다. id는 네이버로부터 부여된 리뷰어의 id이고, document는 실제 텍스트 리뷰이며, label은 0(부정)과 1(긍정)으로 이루어진 감성 클래스 라벨이다.

한국어 감성분석 연구에서 거의 무조건 등장하는 데이터셋이다.

### 2. 영어 감성분석의 대표적인 벤치마크 SST(Stanford Sentiment Treebank)

링크 : https://nlp.stanford.edu/sentiment/

설명 : 약 1만개의 영화 리뷰 문장으로 구성되어있다. 약 21만개의 유니크한 구문이 있다. 스탠포트 parser로 분석되었고, 3명의 심사위원이 주석을 단 215,154개의 parse tree에서 나온 고유한 문구를 갖고있다.

## 3. 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.)

—-

모델명 : **SMART-RoBERTa Large**

논문 명 : SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization

구조 간략 설명

- SMART는 **SM**oothness inducing **A**dversarial **R**egularization and B**R**egman p**R**oximal poin**T** op**T**imization의 약자이다.
- 모델의 복잡성을 효과적으로 관리하는 Smoothness 지향적 정규화를 사용함
- Bregman Proximal Point Optimization 사용 : 모델이 급격하게 업데이트되지 않도록 각 iteration마다 페널티를 부과함.
- RoBERTa(2019)에 기반하고있다.

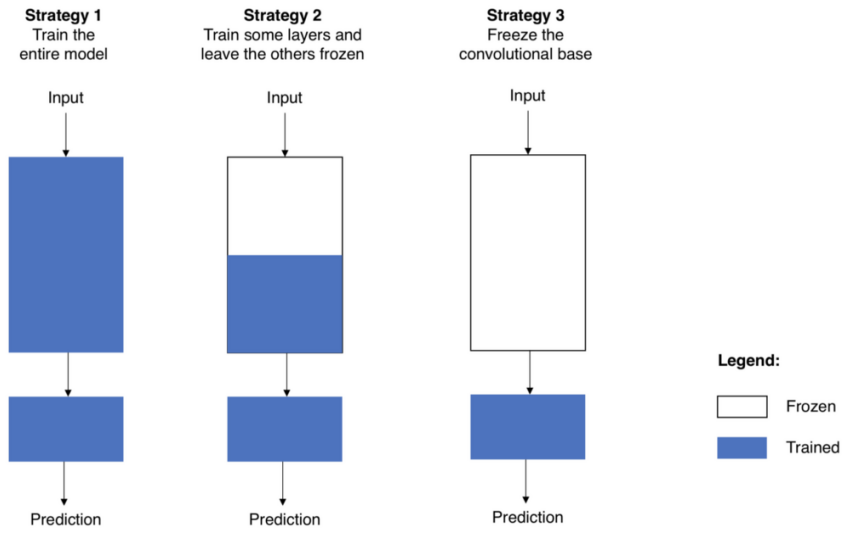
## 4. 학습 방식

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

블로그를 참조하였음

1. 데이터 전처리
2. 사전학습된 모델을 불러옴
   - 파라미터만 가져올 수도 있고, 모델 전체를 불러올 수도 있음
3. 내가 풀고자 하는 task의 문제에 맞게 마지막 층을 classifier 혹은 regressor로 변형함
4. 다음과 같은 전략 중 하나를 선택하여 파라미터를 학습시킴
   - strategy1. 모델 전체를 학습시킴
   - strategy2. 일부 계층만 학습시키고 나머지 계층은 파라미터를 학습시키지 않음

- strategy3. convolutional base는 건들지 않고 classifier만 재학습시킴




---

- (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.
  —-

## 5. 평가 방식

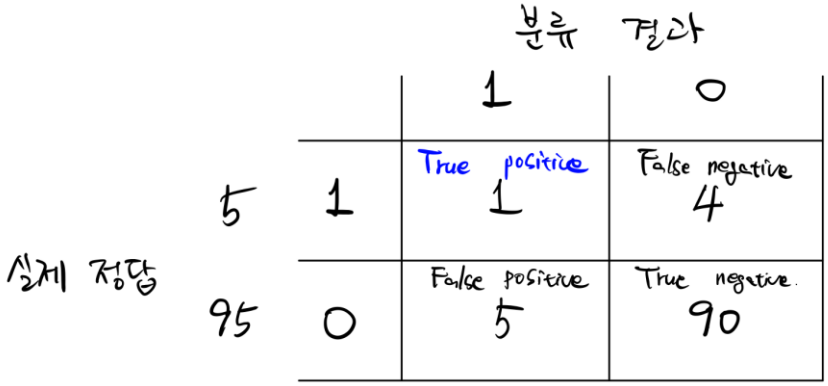
긍부정 예측 task에서 주로 사용하는 평가 지표를 최소 4개 조사하고 설명하세요.
—-

이전에 스스로 정리한 글과 이 페이지를 참조하였음

| Confusion Matrix | | Predicted | |
|---|---|---|---|
| | | 1(+) | 0(-) |
| Actual | 1(+) | $n_{11}$ | $n_{10}$ |
| | 0(-) | $n_{01}$ | $n_{00}$ |

## 1. Accuracy

가장 직관적인 비율이다. 전체 데이터 샘플 중에서 올바르게 분류된 샘플의 수를 비율로 나타낸 것이다.
그런데 Accuracy는 데이터에 따라 잘못된 통계를 나타낼수도 있다. 예를 들어, 스팸메일이면 1, 스팸메일이 아니면 0이라고 예측하는 모델을 만들었다고 해보자. 근데, 실제로는 정상적인 메일이 대다수고 스팸메일은 별로 없다. 만약 정상 메일이 95개, 스팸메일이 5개라고 해보자. 그리고 이 모델이 스팸메일을 1개밖에 걸러내지 못했다고 해보자. 그럼 Confusion matrix는 어떻게 될까?




여기서 Accuracy를 구해보면,96/100=0.96, 무려 96%의 정확도를 가진 모델이 된다. 언뜻 보면 높아보이지만, 높은 성능의 모델이라고 절대 할 수 없다.

$$\text{Accuracy (1 - Misclassification error, 정확도)} = \frac{n_{11} + n_{00}}{n_{11} + n_{10} + n_{01} + n_{00}}$$

## 2. Recall

이를 정확하게 판단하기 위해서 Recall(재현율)이라는 지표가 존재한다. 수식은 다음과 같다.

$$\text{Recall(Sensitivity, 재현율)} = \frac{n_{11}}{n_{11}+n_{10}} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$

즉, Recall은 1을 가지고만 계산한다. 실제 1 중에서 1로 예측한 것이 몇개인지를 본다. 당연하게도, 이런 스팸메일 분류같은 경우에는 Accuracy보다는 Recall을 사용하여 성능을 나타내는것이 합리적이다.

### 3. Precision

근데 Recall은 실제 1 중에서 예측 1이 얼만큼 채워졌는지만을 보기 때문에 싹다 1로 예측해버리면 속수무책이다. 이런 상황에서는 Precision(정밀도)가 도움이 될 수 있다. Pecision은 다음과 같다.

$$\text{Precision (정밀도)} = \frac{n_{11}}{n_{11}+n_{01}} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

예측 1에 대한 실제 1의 비율이다. 예측을 1로 내놓은 것 중에서 실제 1이 얼마나 차지하는지를 나타낸 것이다.

### 4. F1-score

지금까지를 요약해보면 이렇다.
실제 1 중에서 예측 1이 얼마나 차지하는가? → Recall.
예측 1 중에서 실제 1이 얼마나 차지하는가? → Precision.
둘 다 성능을 계산하는 지표로서 타당해보인다.

$$\text{F1 score (hormonized mean of recall and precision)} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

F1-Score는 Recall과 Precision의 조화평균으로 나타낸다.