DKTC task

¿팀명모하조¿

맹선재, 윤상현, 황인준

목차

1. Identifying Task
2. Finding Solution
3. About Data
4. Selecting Model
5. KoBERT
6. KoELECTRA
7. KLUE BERT
8. Method Used
9. Results

Process

1일차

- 일반 대화 데이터 세트 추가
- 데이터 전처리 후 시간이 남으면 모델 만들어 적용해 보기 (GPT 사용?)
- 데이터 전처리에 관한 논의 및 시행착오
- 어떤 모델을 써볼지, 어떤 식으로 조정을 해볼지에 대한 논의 진행

2일차

- 문제점 인지
- 일반 대화가 불필요하다는 것을 인지
- 데이터 전처리 후 모델 작성 및 실험 시작 (KoBERT)
- 성능 미달로 다른 모델 작성 및 실험 시작 (KoBERT-KoELECTRA Ensemble)
- 추가 성능 개선 위해 다른 모델 작성 및 실험 시작 (KoELECTRA-KLUE BERT)
- 시행착오를 거치며 목표 설정 및 다양한 모델로 실험 진행 (목표: 리더보드 1위-달성)

1. Identifying Task

- 입력데이터는 무엇이고, 무엇을 예측하려 하는가?

  문장을 읽고, 대화가 어떤 유형에 속하는지 맞추는 문제

  즉, input : 문장, output : label
  => Classification

- 언어의 속성 : 한국어 _ 특수한 처리가 필요
  띄어쓰기단위로 토큰화 한 뒤, 추가적인 '불용어 처리' 필요

- 추가적인 데이터 확보의 필요성
  '일반 대화' DataSet을 추가했었지만, 삭제함

2. Finding Solution

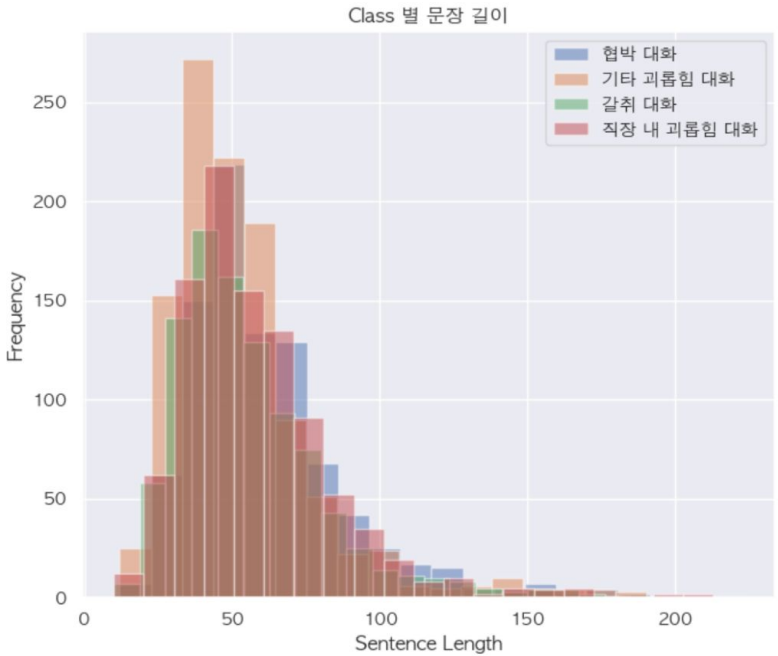
- Classification문제라는 것은 확인하였다.

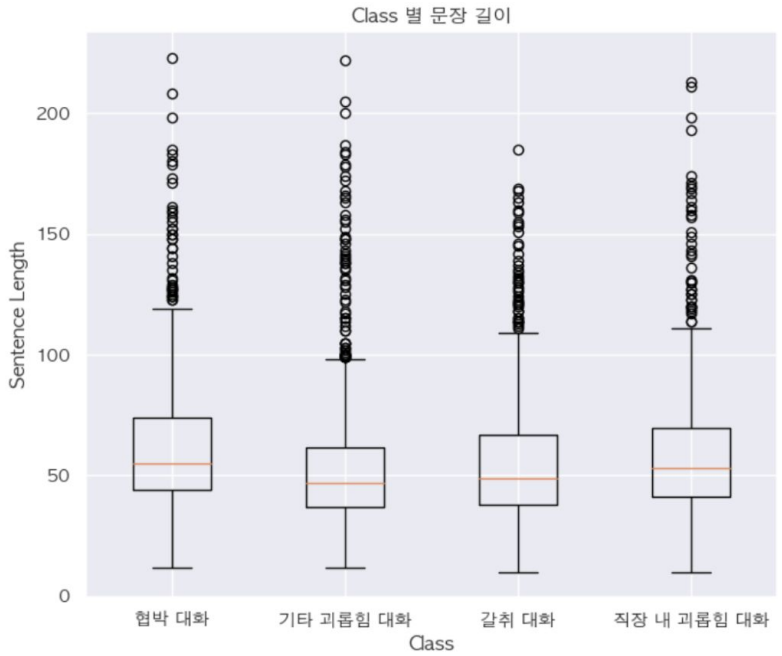
- Incoder vs Decoder?
  incoder는 문장을 이해하는데 높은 성과를 보이고
  Decoder는 문장의 생성에 높은 성과를 보인다.
  (왜 그런지 좀 더 자세하게 추가하면 좋음)

- Incoder문제라면, 기존에 있는 모델로 해결이
  가능한가?
  Yes

- 그렇다면 어떤 모델을 사용할 것인가?
  BERT, ELMo, ELECTRA 중 BERT를 채택

3. About ~~time~~ Data





- Data의 분포가 유사하기 때문에 균형이 맞는 것으로 판단

3. About Data

```
def clean_text(text):

    # 한글, 영어, 숫자, 공백 제외 다 지우기
    text = re.sub(r'[^가-힣a-z0-9\s]', ' ', conversation)

    # 공백 많은거 제거
    text = ' '.join(conversation.split())

    # 불용어 처리
    stopwords = [
      '이', '있', '하', '것', '들', '그', '되', '수', '이', '보', '않', '없', '나', '사람', '주', '아니',
      '등', '같', '우리', '때', '년', '가', '한', '지', '대하', '오', '말', '일', '그렇', '위하',
      '때문', '그것', '두', '말하', '알', '그러나', '받', '못하', '일', '그런', '또', '문제', '더', '사회',
      '많', '그리고', '좋', '크', '따르', '중', '나오', '가지', '씨', '시키', '만들', '지금', '생각하',
      '그러', '속', '하나', '집', '살', '모르', '적', '월', '데', '자신', '안', '어떤', '내', '내', '경우',
      '명', '생각', '시간', '그녀', '다시', '이런', '앞', '보이', '번', '나', '다른', '어떻', '여자', '개',
      '전', '들', '사실', '이렇', '점', '싶', '말', '정도', '좀', '원', '잘', '통하', '소리', '놓'
    ]

    return text

test_data['conversation'] = test_data['conversation'].apply(clean_text)
```

4. Selecting Model

Q) RNN, LSTM, S2S은 어떨까?

- Data의 문장 길이를 봤을 때, RNN과 LSTM은 Vanishing Gradients문제가 발생할 수 있다고 판단.
- S2S는 ContextVector를 사용하기 때문에, 고정된 크기로 전달하는데 한계가 발생 & BottleNeck

Q) 왜 BERT를 사용했을까?

- 먼저 Classification에는 encoder 구조가 유리
- ELMo, ELECTRA보다는 BERT의 Bidirectional이 유효하게 작용할 것이라 생각

5. Selecting Model-Ko-BERT

BERT: 구글이 제안한 모델

-사전 학습된 대용량의 레이블링 되지 않는 데이터를 이용하여 언어 모델을 학습하고 이를 토대로 특정 작업을 위한 신경망을 추가하는 전이 학습 방법

-기본적으로 대량의 단어 임베딩 등에 대해 사전 학습이 되어 있는 모델을 제공하기 때문에 상대적으로 적은 자원만으로 여러 task 수행 가능

SKT에서 공개한 위키피디아, 뉴스 등에서 수집한 5천만개의 문장으로 학습된 모델

한국어의 불규칙한 언어 변화의 특성을 반영하기 위해 데이터 기반 토큰화 (SentencePiece tokenizer) 기법을 적용

vocab 크기 8002, 모델 파라미터 크기 92M

6. Selecting Model-Ko-ELECTRA

ELECTRA: generator에서 나온 token을 보고 discriminator에서 "real" token인지 "fake" token인지 판별하는 방법으로 학습

-모든 input token에 대해 학습할 수 있다는 장점을 가지며, BERT 등과 비교했을 때 더 좋은 성능을 보임

34GB의 한국어 text로 학습

Wordpiece 사용, 모델 s3 업로드 등을 통해 OS 상관없이 Transformers 라이브러리만 설치하면 곧바로 사용할 수 있음

7. KoELECTRA

1. 50+회 실험 (후반부에는 모델이 최대 성능-ACC 0.9025-내는 상태에서 고정)
2. BERT에 비해 매우 높은 성적 (0.6대에 그치던 BERT에 반해 시작부터 0.8대)
3. 일부의 토큰을 사전 학습에만 활용하는 BERT와 달리 모든 토큰을 사전 학습과 Fine-tuning 모두에서 활용
4. 추측: 작은 Train과 Test 데이터셋의 토큰 모두를 학습하기에 더 좋은 성능을 보였음?

8. Selecting Model-KLUE-BERT

-벤치마크 데이터인 KLUE에서 베이스라인으로 사용되었던 모델

-모두의 말뭉치, CC-100-Kor, 나무위키, 뉴스, 청원 등 문서에서 추출한 63GB의 데이터로 학습

-Morpheme-based Subword Tokenizer

-vocab size 32,000, 모델 파라미터 크기 111M

9. Method Used-Ensemble

-여러 개의 개별 모델을 조합하여 최적의 모델로 일반화하는 방법

-일반적으로 보팅, 배깅, 부스팅 세 가지의 유형으로 나뉨

-태스크에 사용한 방식은 소프트 보팅 (모델 별 예측 확률값의 평균)

보팅 (Voting)

-여러 개의 분류기가 투표를 통해 최종 예측 결과를 결정

-서로 다른 알고리즘을 여러 개 결합하여 사용눌 수 있음

배깅 (Bagging)

-데이터 샘플링을 통해 모델을 학습시키고 결과를 집계

-모두 같은 유형의 알고리즘 기반의 분류기를 사용

-데이터 분할 시 중복을 허용

-과적합 방지에 효과적

부스팅 (Boosting)

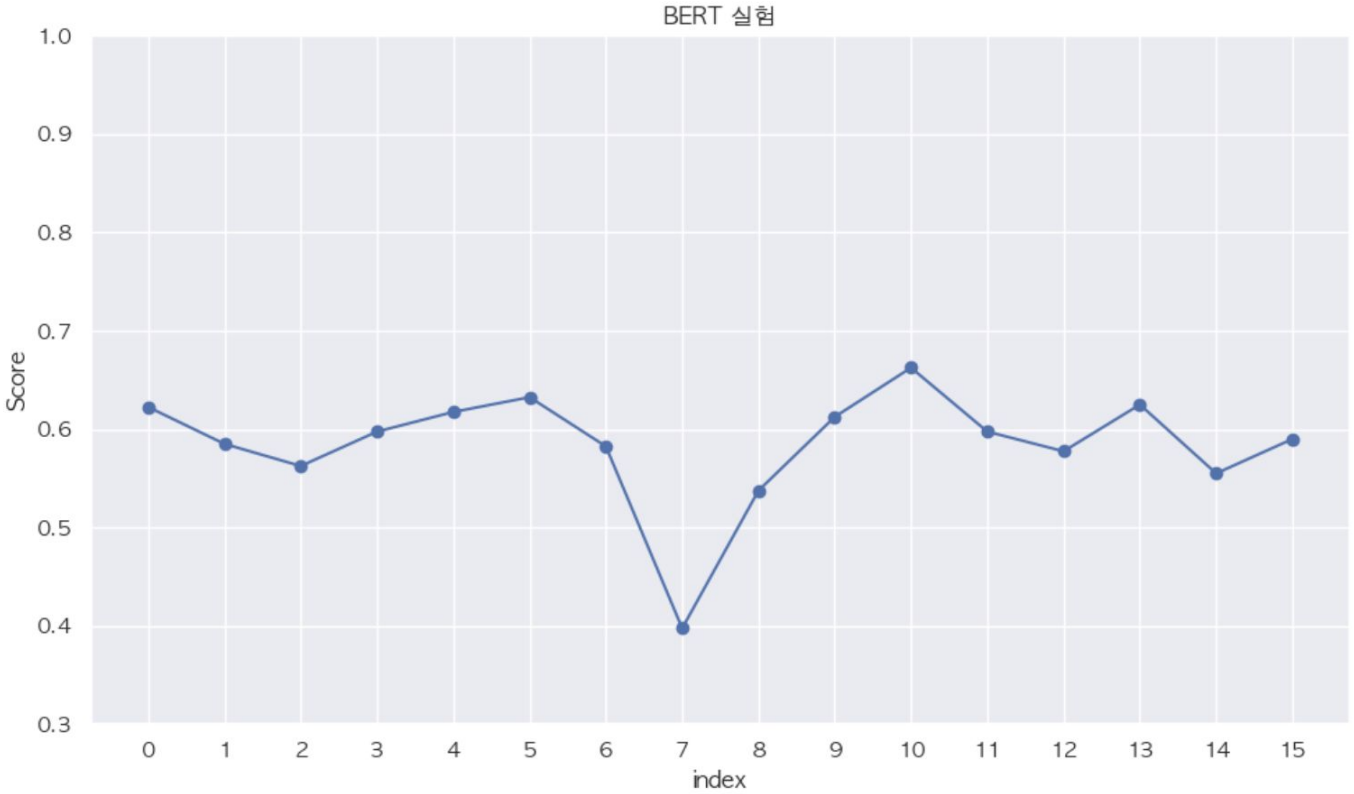
-이전 분류기가 예측이 틀린 데이터에 대해서 올바르게 예측할 수 있도록 다음 분류기에게 가중치를 부여하면서 순차적으로 학습과 예측을 진행

-예측 성능이 뛰어나 앙상블 학습을 주도

-보통 배깅에 비해 성능이 좋지만, 속도가 느리고 과적합이 발생할 가능성이 존재

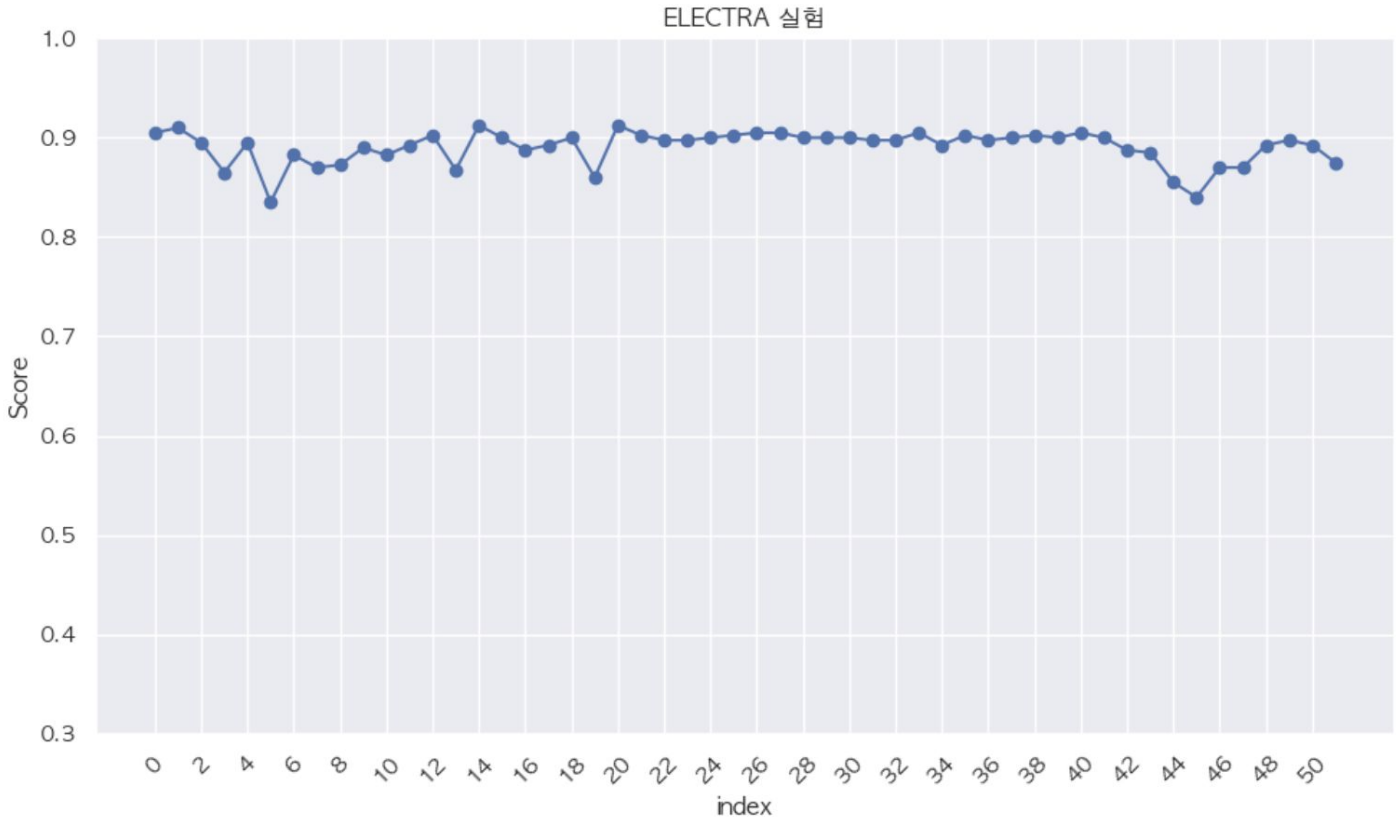
Results

1. KoBERT

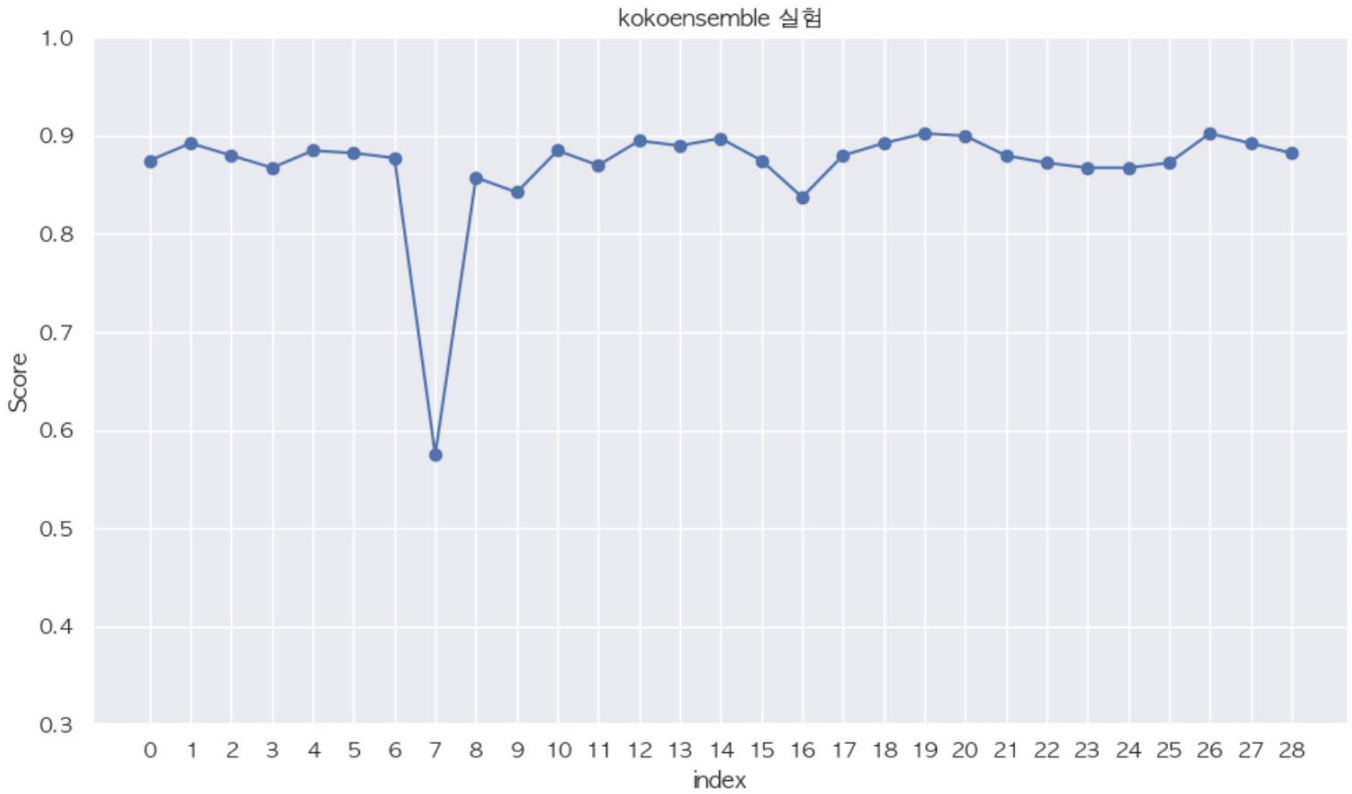


1. 12번의 실험
2. 리더보드에는 0.7 이상의 성능이 존재했음
3. 생각보다 낮은 성능을 기록

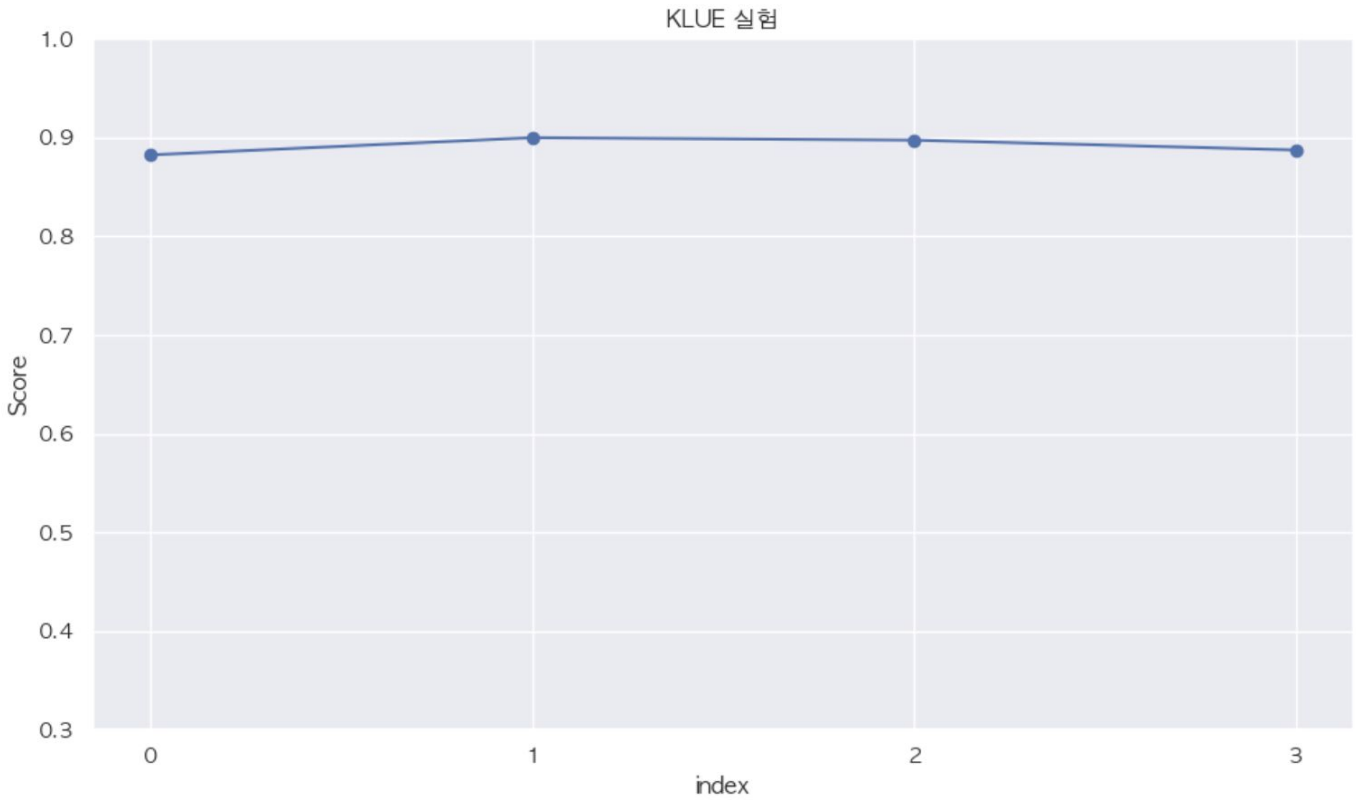
2. KoELECTRA


ELECTRA 실험
Score
index
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

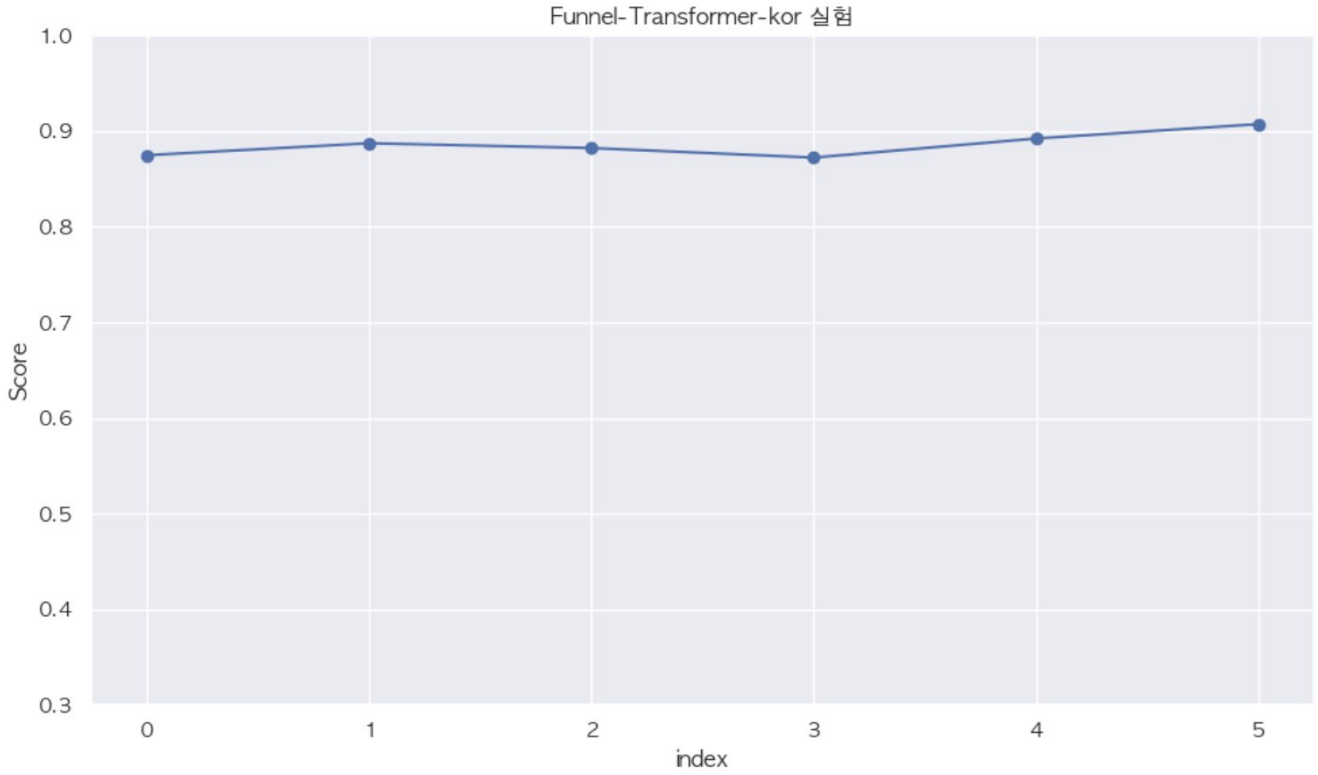
3. KoKo-Ensemble (KoBert + KoELECTRA)



4. KLUE BERT



5. Funnel-Transformer-Kor



6. Leader Board

ELECTRA & KLUE & Funnel-Transformer를 soft voting으로 Ensemble

Score : 0.92로 1위

[팀명모하조] Ensemble is all you need 0.92

정답TEST 1

Rubric

평가문항	상세기준
1. 데이터 전처리 과정이 잘 이루어졌는가?	데이터 정제 및 토큰화를 잘 수행하였다. 모델 학습에 충분한 데이터를 추가 수집 및 augmentation 하였다.
2. 적절한 모델 및 평가지표를 선정했는가?	한국어 문장 분류에 적절한 모델과, 해당 데이터셋 분류에 적합한 metric을 선택하고 선택 근거 및 결과를 잘 분석하였다.
3. 모델의 분류 성능을 얼마나 달성했는가?	선정한 metric에 따라 성능 개선 과정을 분석하고 목표한 수치 이상의 성능을 달성해냈다.

Self-Check

- 데이터 EDA와 데이터 전처리가 적절하게 이뤄졌는가? -△
- Task에 알맞게 적절한 모델을 찾아보고 선정했는가? -o
- 성능향상을 위해 논리적으로 접근했는가? -o
- 결과 도출을 위해 여러가지 시도를 진행했는가? - o
- 도출된 결론에 충분한 설득력이 있는가? -o
- 적절한 metric을 설정하고 그 사용 근거 및 결과를 분석하였는가? -△

모델 공유

klue-bert 성능 0.9125

https://drive.google.com/file/d/1UWeCmYoqbU0qlXdf-KWpLUyHZg-bgvRD/view?usp=sharing

ko-electra 성능 0.8975

https://drive.google.com/file/d/10fKZhuSTNkj5UZMV_mQbE_6cZgY5lYbz/view?usp=drive_link

funnel-transformer-kor 성능 0.9075
https://drive.google.com/file/d/1k2iMVVmxinmFTv1DgRxLY67A0un2XeNB/view?usp=sharing

Ref

kobert: https://huggingface.co/monologg/kobert

klue: https://huggingface.co/klue/bert-base

koelectra: https://huggingface.co/monologg/koelectra-base-v3-discriminator

funnel-kor-transformer: https://huggingface.co/kykim/funnel-kor-base