DACON

SUNG KYUN KWAN UNIVERSITY 1398 성균관대학교

문장 유형 분류 AI 경진대회 1위

제출수늘려주세요

김광륜

PyTorch

발표 순서

1. 대회목표와 데이터 설명
2. 성능 개선을 위한 시도
    2.1. Regex
    2.2. Augmentation
    2.3. Loss
    2.4. Ensemble
    2.5. Custom Model
3. 개발환경
4. 출처
5. 결론 및 의견

대회 목표 및 데이터

[주제]

문장 유형 분류 AI 모델 개발

[설명]

언어가 사용되는 모든 영역에서 폭넓게 활용될 수 있는 문장 유형 분류 AI 모델을 개발해 주세요.

문장을 입력으로 받아 문장의 '유형', '시제', '극성', '확실성'을 AI 분류 모델 생성

[주최 / 주관]

주최: 성균관대학교
주관: 데이콘

대회 목표 및 데이터

- train.csv

	문장	유형	극성	시제	확실성	label
0	0.75%포인트 금리 인상은 1994년 이후 28년 만에 처음이다.	사실형	긍정	현재	확실	사실형-긍정-현재-확실
1	이어 " 앞으로 전문가들과 함께 4주 단위로 상황을 재평가할 예정 " 이라며 " 그 이전이...	사실형	긍정	과거	확실	사실형-긍정-과거-확실
2	정부가 고유가 대응을 위해 7월부터 연말까지 유류세 인하 폭을 30%에서 37%까지...	사실형	긍정	미래	확실	사실형-긍정-미래-확실
3	서울시는 올해 3월 즉시 견인 유예시간 60분을 제공하겠다고 밝혔지만, 하루 만에 ...	사실형	긍정	과거	확실	사실형-긍정-과거-확실
4	익사한 자는 사다리에 태워 거꾸로 놓고 소금으로 코를 막아 가득 채운다.	사실형	긍정	현재	확실	사실형-긍정-현재-확실
...	...	...	...	...	...	...
16536	' 신동멈 ' 은 ' 신비한 동물사전 ' 과 ' 해리 포터 ' 시리즈를 잇는 마법 어드벤처물로, ...	사실형	긍정	과거	확실	사실형-긍정-과거-확실
16537	수족냉증은 어릴 때부터 심했으며 관절은 어디 한 곳이 아니고 목, 어깨, 팔꿈치, ...	사실형	긍정	과거	확실	사실형-긍정-과거-확실
16538	김금희 소설가는 " 계약서 조정이 그리 어려운가 작가를 격려한다면서 그런 문구 하나 ...	사실형	긍정	과거	확실	사실형-긍정-과거-확실
16539	1만명이 넘는 방문자수를 기록한 이번 전시회는 총 77개 작품을 넥슨 사옥을 그대로...	사실형	긍정	과거	불확실	사실형-긍정-과거-불확실
16540	《목민심서》의 내용이다.	사실형	긍정	현재	확실	사실형-긍정-현재-확실

16541 rows × 6 columns

```
(array(['사실형', '추론형', '예측형', '대화형'], dtype=object),
 array(['긍정', '부정', '미정'], dtype=object),
 array(['현재', '과거', '미래'], dtype=object),
 array(['확실', '불확실'], dtype=object))
```

대회 목표 및 데이터

- 예측값(label)

유형(4개), 극성(3개), 시제(3개), 확실성(2개)

각 4개의 feature 조합

유형	극성	시제	확실성	label
사실형	긍정	현재	확실	사실형-긍정-현재-확실
사실형	긍정	과거	확실	사실형-긍정-과거-확실
사실형	긍정	미래	확실	사실형-긍정-미래-확실
사실형	긍정	과거	확실	사실형-긍정-과거-확실
사실형	긍정	현재	확실	사실형-긍정-현재-확실
...	...	...	...	...
사실형	긍정	과거	확실	사실형-긍정-과거-확실
사실형	긍정	과거	확실	사실형-긍정-과거-확실
사실형	긍정	과거	확실	사실형-긍정-과거-확실
사실형	긍정	과거	불확실	사실형-긍정-과거-불확실
사실형	긍정	현재	확실	사실형-긍정-현재-확실

성능을 올리기 위한 시도

1. **Text Augmentation**
   - replaced 2 words in one sentence and made 2 or 3 sentences.(randomly)
2. **Ensemble**
3. **Customize model layers**
   - Residual Blocks
4. **Change loss function**
   - Focal Loss
   - Asymmetric Loss

   (https://github.com/Alibaba-MIIL/ASL/blob/main/src/loss_functions/losses.py)

   (https://paperswithcode.com/paper/asymmetric-loss-for-multi-label)

5. **Fold(just 5 folds)**
6. **Huggingface Custom Trainer** 🤗
7. **Custom Scheduler (안썼습니다.)**

Preprocessing

- Regex

특수기호와 숫자 한글 등이 섞여있어

일괄적용을 통해 정규화의 효과를 기대하기 위해 진행

유형과 시제 Feature에 숫자가 영향을 끼칠 것 같아 숫자는 유지

2회 이상의 띄어쓰기 1개로 적용

```
train['문장'] = train['문장'].apply(lambda x: re.sub("[^ A-Za-z0-9가-힣]", "", x))
train['문장'] = train['문장'].apply(lambda x: re.sub("[ +]", " ", x))

test['문장'] = test['문장'].apply(lambda x: re.sub("[^ A-Za-z0-9가-힣]", "", x))
test['문장'] = test['문장'].apply(lambda x: re.sub("[ +]", " ", x))
```

Augmentation

- RS 방식

입력된 문장을 띄어쓰기 별로 단어를 분할 후 무작위 2 단어를 선정하여 두 단어의 위치를 교체하는 방식을 채택

```
origin :075포인트 금리 인상은 1994년 이후 28년 만에 처음이다

Augmentaion :075포인트 금리 인상은 이후 1994년 28년 만에 처음이다

Augmentaion :금리 075포인트 인상은 1994년 이후 28년 만에 처음이다

Augmentaion :처음이다 금리 인상은 1994년 이후 28년 만에 075포인트
```

다양한 Text 증강방식이 존재하지만, 대회의 label 특성상 임의로 단어를 교체하거나 삭제하게 되면 label 예측에 큰 영향이 미칠 거란 판단하에 최대한 보수적인 증강 방식을 적용

출처 : https://github.com/catSirup/KorEDA/blob/master/eda.py

Augmentation

	문장	유형	극성	시제	확실성	label
0	전국 더욱 뽑는 비례대표 국회의원이 따로 규정된 것까지 감안하면 국회의원이 국민 대...	사실형	긍정	현재	확실	사실형-긍정-현재-확실
1	우선 최근 소비자들 있는 재조명되고 사이에서 모델은 기본형에 해당하는 S20이다	사실형	긍정	현재	확실	사실형-긍정-현재-확실
2	무협 관계자는 안전운임도 부담이 나날이 증가하지만 기업들은 할증으로 운동장이어서 부...	사실형	부정	과거	확실	사실형-부정-과거-확실
3	그는 2020년까지 남북한이 통일되지 않을 경우를 상상하기는 매우 어렵다고 했다	사실형	부정	과거	확실	사실형-부정-과거-확실
4	요 몇 년 옛 할리우드에서 새 히트작이 자꾸 리메이크된 것은 신박한 이야기의 고갈을...	사실형	긍정	과거	확실	사실형-긍정-과거-확실
...	...	...	...	...	...	...
64667	노인일자리 비판은 60만명이 넘는 빈곤노인을 그냥 방치하자는 것이냐	대화형	긍정	현재	불확실	대화형-긍정-현재-불확실
64668	트위터 역사상 가장 높은 빈도로 공유된 이 뉴스들은 합병이 끝나자 바로 사라졌다	사실형	긍정	과거	확실	사실형-긍정-과거-확실
64669	우곡은 다시 태어나더라도 누룩 개발에 전념하겠다는 뜻으로 평생을 누룩과 술 개발에 ...	사실형	긍정	현재	확실	사실형-긍정-현재-확실
64670	고대일록의 저자 정경운도 전염병에 딸을 잃는다	사실형	긍정	현재	확실	사실형-긍정-현재-확실
64671	이것도 생필품까지 모자라 갈취해갔다	사실형	긍정	과거	확실	사실형-긍정-과거-확실

64672 rows × 6 columns

```
train['문장'].str.len().max(), test['문장'].str.len().max()
```

```
(496, 378)
```

출처 : *https://github.com/catSirup/KorEDA/blob/master/eda.py*

Loss

대회의 데이터는 Imbalance 문제를 겪고, 보통은 upsampling을 쓰는 smote 기법이나 undersampling 등을 많이 사용하지만 크게 효과를 내지못해 Loss로 해결하고자 탐색

- Focal Loss
- Asymmetric Loss

Focal Loss

- Cross Entropy의 클래스 불균형 문제를 다루기 위한 개선된 버전이라고 말할 수 있으며 어렵거나 쉽게 오분류되는 케이스에 대하여 더 큰 가중치를 주는 방법을 사용합니다. 반대로 쉬운 케이스의 경우 낮은 가중치를 반영합니다.

출처 : https://gaussian37.github.io/dl-concept-focal_loss/

Asymmetric Loss

- Focal Loss는 Positive, Negative 모두 같은 값으로 적용하는 문제점이 존재
- ASL로 negative-positive imbalance, mislabeling 해결, 기존 아키텍쳐 구조를 변경하지 않기 때문에 모델 학습시간이나 추론 시간의 부하는 없으면서 성능이 좋다
- ***기존 모델을 Single Label로 설계했기 때문에 시간 관계상 Single Label ASL을 적용(sum loss)***

출처 : https://github.com/Alibaba-MIIL/ASL/blob/main/src/loss_functions/losses.py

앙상블 경로

- 740
  - .ipynb
- 741
  - .ipynb
- 743
  - .ipynb
- 744
  - .ipynb
- 749
  - .ipynb
- 7474
  - .ipynb
- 앙상블.ipynb

각 폴더의 숫자는 추론을 제출했을때의 Public Score 입니다.

앙상블 모델 정보

- 공통사항 5 KFold
- 740
  - kykim/electra-kor-base
  - 3 Augmentation
  - Focal loss
- 741
  - monologg/kobigbird-bert-base / Custom Layer
  - 3 Augmentation
  - ASLoss
- 743
  - beomi/KcELECTRA-base-v2022
  - 3 Augmentation
  - Focal loss
- 744
  - monologg/kobigbird-bert-base
  - 3 Augmentation
  - ASLoss
- 7474
  - monologg/kobigbird-bert-base
  - 2 Augmentation
  - Focal loss
- 749
  - monologg/kobigbird-bert-base
  - 3 Augmentation
  - Focal loss

741 Custom Model

- 각 Feature의 logit이 서로 영향을 미칠 것이라는 가설을세워 접근한 Residual
- out4
- out4 + out3
- out4 + out3 + out2
- out4 + out3 + out2 + out1

```python
class CustomModel(nn.Module):
    def __init__(self):
        super(CustomModel, self).__init__()
        if model_path == 'monologg/kobigbird-bert-base':
            config.attention_type = "original_full"
        self.base_model = AutoModel.from_pretrained(model_path, config=config)
        self.out = self.base_model.encoder.layer[-1].output.dense.out_features//2
        self.norm = 384

        self.Linear1 = nn.Sequential(
            nn.Linear(768, 384),
            nn.BatchNorm1d(self.norm))
        self.Linear2 = nn.Sequential(
            nn.Linear(768, 384),
            nn.BatchNorm1d(self.norm))
        self.Linear3 = nn.Sequential(
            nn.Linear(768, 384),
            nn.BatchNorm1d(self.norm))
        self.Linear4 = nn.Sequential(
            nn.Linear(768, 384),
            nn.BatchNorm1d(self.norm))

        self.type_classifier = nn.Sequential(
            nn.Dropout(p=0.2),
            nn.Linear(in_features=self.out, out_features=4),
        )
        self.polarity_classifier = nn.Sequential(
            nn.Dropout(p=0.2),
            nn.Linear(in_features=self.out, out_features=3),
        )
        self.tense_classifier = nn.Sequential(
            nn.Dropout(p=0.2),
            nn.Linear(in_features=self.out, out_features=3),
        )
        self.certainty_classifier = nn.Sequential(
            nn.Dropout(p=0.2),
            nn.Linear(in_features=self.out, out_features=2),
        )

    def forward(self, input_ids, attention_mask, labels=None, token_type_ids=None):
        x = self.base_model(input_ids=input_ids, attention_mask=attention_mask)[0]
        x = x[:,0,:]

        out4 = self.Linear4(x)
        certainty_output = self.certainty_classifier(out4)

        out3 = self.Linear3(x)
        tense_output = self.tense_classifier((out3+out4))

        out2 = self.Linear2(x)
        polarity_output = self.polarity_classifier((out2+out3+out4))

        out1 = self.Linear1(x)
        type_output = self.type_classifier((out1+out2+out3+out4))

        return type_output, polarity_output, tense_output, certainty_output
```

개발환경

OS

- windows 10
- ubuntu 22.04
- aws ec2 p3 2xlarge

라이브러리

- torch 1.13.0
- cuda 11.7.1

사용한 사전학습모델과 출처

- **kykim/electra-kor-base**

  (https://github.com/kiyoungkim1/LMkor)

  (https://huggingface.co/kykim/electra-kor-base)

  (https://openreview.net/pdf?id=r1xMH1BtvB)
- **monologg/kobigbird-bert-base**

  (https://github.com/monologg/KoBigBird)

  (https://huggingface.co/monologg/kobigbird-bert-base)

  (https://arxiv.org/abs/2007.14062)
- **beomi/KcELECTRA-base-v2022**

  (https://github.com/Beomi/KcELECTRA)

  (https://huggingface.co/beomi/KcELECTRA-base-v2022)

  (https://openreview.net/pdf?id=r1xMH1BtvB)

결론 및 의견

결론

- 효과적이었던 방식은 Ensemble > Fold > Augmentation 순이었다.
- Ensemble은 완전히 다른 구조의 모델에서 효과적이었다.
- 비효과적이었던 방식은 Seed Ensemble, 영번역모델이었다.

의견

- Imbalanced 한 데이터에서 사용되는 기술
  예) Sampling/Loss를 적용하여 검증하는 접근
- 번역후 모델적용시 성능이 매우 낮았는데, 이를 통한 새로운 접근이 가능할지
  이와 관련하여 한글 Pretrained model의 한계가 존재하므로 한글의 Pretrained model 관련 연구 및 지원의 필요성 예) SOTA Deberta

감사합니다

제출수늘려주세요

김광륜