# 웰니스 데이터셋 상세 분석

## 목차

*Note*

# GPU 사용 Test

***Note***

***지난 발표에서, 지원받은 서버의 GPU 사용 여부 테스트를 위해 KoBERT 모델을 이용한 카테고리 분류 진행***

```
100% ████████████████████ 66/66 [00:04<00:00, 14.54it/s]

epoch 4 batch id 1 loss 3.489006519317627 train acc 0.375
epoch 4 train acc 0.4097222222222227

100% ████████████████████ 17/17 [00:00<00:00, 51.31it/s]

epoch 4 val acc 0.38566975703324813

100% ████████████████████ 66/66 [00:04<00:00, 14.52it/s]

epoch 5 batch id 1 loss 3.1293299198150635 train acc 0.375
epoch 5 train acc 0.4737215909090909

100% ████████████████████ 17/17 [00:00<00:00, 51.08it/s]

epoch 5 val acc 0.40221387468030695
```

```
(base) ubuntu@nipa2022-65543:~$ nvidia-smi
Mon Aug  1 13:15:03 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.106.00   Driver Version: 460.106.00   CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  A100-PCIE-40GB      On   | 00000000:00:05.0 Off |                    Off |
| N/A   45C    P0   203W / 250W |   9084MiB / 40536MiB |     83%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A     14865      C   ...3/envs/chatbot/bin/python     9081MiB |
+-----------------------------------------------------------------------------+
```

**-> 성능이 매우 좋지 않은 것을 확인할 수 있으며, 이는 크게 세가지 원인 - 데이터셋 부족, 많은 개수의 카테고리 (173개), 모델 구조 – 로 보임**

# 우울증 관련 카테고리 선정

***Note***

## *카테고리를 조정해야 할 필요성을 느끼고, Wellness_Conversation_02 데이터셋을 사용하기로 함*

| | 핵심증상 | intent | keyword(임상 키워드) | 특이 사항 | 연관표현 | utterance | utterance(2차) | response(공감) | 임상질문그룹(연세의료원제공) | utterance(긍정) | utterance(부정) | 긍정에 대한 챗봇 답변 | 부정에 대한 챗봇 답변 | 추가발화 (190917) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 지속되는 우울한기분(우울감) | 정신증상/우울감 | 우울 | NaN | 머리가 짓눌러지는 느낌/머리가 맑지 않다/침울하면서 잠도 못 자다/후회가 많고 침울... | 우울해 | 임신해서 우울해 | 기분이 우울하시군요. 00님에게 스트레스 받는 일이 있었던 건 아닌지 걱정스러워요. | 거의 매일, 하루 중 대부분 시간을 우울한 기분으로 있거나 가라앉았던 적이 있나요? | 응 | 아니 | 저런... (우는 이모티콘) 기분이 나아지지 않는 상태군요. | 그렇군요. 기분이 계속 처지신다면 편안한 음악을 한번 들어보는 것도 좋을 것 같아요. | NaN |
| 1 | NaN | NaN | NaN | NaN | NaN | 너무 우울해 | 아이 가지고 나서 우울해 | 마음이 우울해서 힘드셨군요. 00님의 소식이 궁금했는데 마침 저를 잘 찾아주셨네요. | NaN | 어 | 아닌데? | 힘든 상황이시군요. 저랑 더 얘기를 나눠볼까요? | 그랬군요. 또 기분이 안 좋아지면 언제든 저에게 이야기해주세요. | NaN |
| 2 | NaN | NaN | NaN | NaN | NaN | 우울해죽겠어 | 아이 가졌는데 기분 하나도 안 좋고 울적해 | 00님이 울적하시다니 저도 마음이 안 좋아요. | NaN | 엉 | 노 | 한번 가라앉은 기분을 다시 돌리기는 참 어려운 것 같아요. 저와 계속 얘기해봐요. | 그런 증상은 없으셨군요. 00님의 머릿속에 낀 먹구름이 얼른 걷혔으면 좋겠어요. | NaN |
| 3 | NaN | NaN | NaN | NaN | NaN | 기분이 우울해 | 임신했는데 남편이 하나도 안 챙겨줘서 우울하다 | 얼굴에서 00님의 기분이 가라앉은 것이 느껴져요. 큰일이 있으신 건 아니었으면 좋겠어요. | NaN | 예스 | 아니거든 | NaN | NaN | NaN |
| 4 | NaN | NaN | NaN | NaN | NaN | 우울모드임 | 진단 결과 안 좋게 나올 것 같아서 우울해 | NaN | NaN | 네 | 아니야 | NaN | NaN | NaN |

**-> 이 중 intent만 따로 뽑아서 보면 다음 값들 보유**

```
(20, array(['정신증상/우울감', nan, '정신증상/슬픔', '정신증상/외로움', '정신증상/분노', '정신증상/무기력',
       '정신증상/감정조절이상', '정신증상/상실감', '정신증상/식욕저하', '정신증상/식욕증가', '정신증상/불면',
       '정신증상/초조함', '정신증상/피로', '정신증상/죄책감', '정신증상/집중력저하', '정신증상/자신감저하',
       '정신증상/자존감저하', '정신증상/절망감', '정신증상/자살충동', '정신증상/불안'], dtype=object))
```

# 우울증 관련 카테고리 선정

*Note*

## *Intent 컬럼을 이용한 우울증 분류하기*

```
(20, array(['정신증상/우울감', nan, '정신증상/슬픔', '정신증상/외로움', '정신증상/분노', '정신증상/무기력',
       '정신증상/감정조절이상', '정신증상/상실감', '정신증상/식욕저하', '정신증상/식욕증가', '정신증상/불면',
       '정신증상/초조함', '정신증상/피로', '정신증상/죄책감', '정신증상/집중력저하', '정신증상/자신감저하',
       '정신증상/자존감저하', '정신증상/절망감', '정신증상/자살충동', '정신증상/불안'], dtype=object))
```

**-> 이 중 nan 값을 제외한 각 컬럼들의 인덱스를 추출하고, 레이블링이 되어 있지 않은 데이터들에 대해서도 동일하게 레이블링**

```
data[data['intent'].notna()].index
```

```
Int64Index([    0,  1657,  3446,  4222,  5305,  6593,  6862,  7082,  8177,
             8610, 10517, 13214, 14459, 15683, 15928, 16248, 17475, 18197,
            18844],
           dtype='int64')
```

| | intent | utterance | utterance(2차) | intent_label |
|---|---|---|---|---|
| 0 | 우울감 | 우울해 | 임신해서 우울해 | 0 |
| 1 | 우울감 | 너무 우울해 | 아이 가지고 나서 우울해 | 0 |
| 2 | 우울감 | 우울해죽겠어 | 아이 가졌는데 기분 하나도 안 좋고 울적해 | 0 |
| 3 | 우울감 | 기분이 우울해 | 임신했는데 남편이 하나도 안 챙겨줘서 우울하다 | 0 |
| 4 | 우울감 | 우울모드임 | 진단 결과 안 좋게 나올 것 같아서 우울해 | 0 |
| ... | ... | ... | ... | ... |
| 19764 | 불안 | NaN | 그래도 잠못자고 불안한건 여전해요. | 18 |
| 19765 | 불안 | NaN | 불안함에 항상 시달리니까 잠도 못잤어요. | 18 |
| 19766 | 불안 | NaN | 불안하고 초조해서 잠이 안 와. | 18 |
| 19767 | 불안 | NaN | 너무 불안하니까 밤만 되면 잠이 안 오고 너무 초조해. | 18 |
| 19768 | 불안 | NaN | 불안해서 그런지 요즘 잠도 잘 안 와서 너무 힘들어요. | 18 |

```
array(['우울감', '슬픔', '외로움', '분노', '무기력', '감정조절이상', '상실감', '식욕저하', '식욕증가',
       '불면', '초조함', '피로', '죄책감', '집중력저하', '자신감저하', '자존감저하', '절망감', '자살충동',
       '불안'], dtype=object)
```

# 우울증 관련 카테고리 선정

***Note***

## *각 Intent 별 데이터 개수 조회*

| intent | label_count |
|---|---|
| 감정조절이상 | 269 |
| 무기력 | 1288 |
| 분노 | 1083 |
| 불면 | 1907 |
| 불안 | 925 |
| 상실감 | 220 |
| 슬픔 | 1789 |
| 식욕저하 | 1095 |
| 식욕증가 | 433 |
| 외로움 | 776 |
| 우울감 | 1657 |
| 자살충동 | 647 |
| 자신감저하 | 320 |
| 자존감저하 | 1227 |
| 절망감 | 722 |
| 죄책감 | 1224 |
| 집중력저하 | 245 |
| 초조함 | 2697 |
| 피로 | 1245 |

**초조함이 2,697개로 가장 많으며, 감정조절이상이 269개로 가장 적은 것을 볼 수 있음.**

**카테고리 간에 불균형 문제를 해결하기 위해서는 이후 다른 데이터셋에서 관련 카테고리에**

**해당하는 텍스트를 얻어서 보강하거나, 크롤링을 통해 관련 텍스트를 수집해서 보강하는 것이 필요해보임**

## *BERT 모델 + 학습 파라미터*

```
bertmodel, vocab = get_pytorch_kobert_model(cachedir=".cache")   # BERT 모델 가져오기
```

```
using cached model. /home/ubuntu/chatbot/code/.cache/kobert_v1.zip
using cached model. /home/ubuntu/chatbot/code/.cache/kobert_news_wiki_ko_cased-1087f8699e.spiece
```

```
# BERT Dataset 클래스 생성
class BERTDataset(Dataset):
    def __init__(self, dataset, sent_idx, label_idx, bert_tokenizer, max_len,
                 pad, pair):
        transform = nlp.data.BERTSentenceTransform(
            bert_tokenizer, max_seq_length=max_len, pad=pad, pair=pair)

        self.sentences = [transform([i[sent_idx]]) for i in dataset]
        self.labels = [np.int32(i[label_idx]) for i in dataset]

    def __getitem__(self, i):
        return (self.sentences[i] + (self.labels[i], ))

    def __len__(self):
        return (len(self.labels))
```

```
# 파라미터 설정
max_len = 64     # 토큰 최대 길이
batch_size = 64   # 배치 사이즈
warmup_ratio = 0.1   # 웜-업 비율
num_epochs = 10   # 학습 수
max_grad_norm = 1   # gradient 정규화 최대값
log_interval = 200   # interval 간격
learning_rate =  5e-5   # 학습률
```

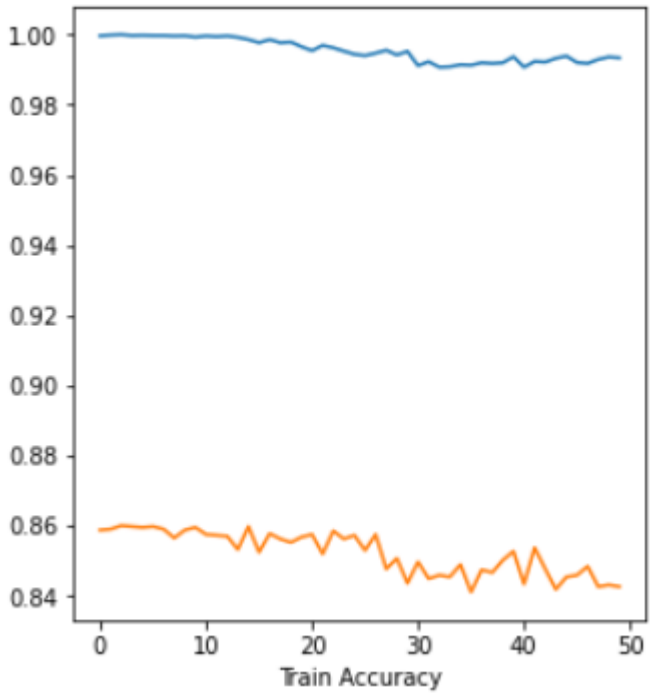
## *BERT 모델 + 학습 파라미터*

```
# BERT Classifier 클래스 생성
class BERTClassifier(nn.Module):
    def __init__(self,
                 bert,
                 hidden_size = 768,
                 num_classes=len(intent_list),
                 dr_rate=None,
                 params=None):
        super(BERTClassifier, self).__init__()    # 부모 클래스 생성자 초기화
        self.bert = bert
        self.dr_rate = dr_rate

        self.classifier = nn.Linear(hidden_size , num_classes)   # 선형 분류기 생성
        if dr_rate:   # 드랍아웃
            self.dropout = nn.Dropout(p=dr_rate)

    def gen_attention_mask(self, token_ids, valid_length):
        attention_mask = torch.zeros_like(token_ids)
        for i, v in enumerate(valid_length):
            attention_mask[i][:v] = 1
        return attention_mask.float()

    def forward(self, token_ids, valid_length, segment_ids):
        attention_mask = self.gen_attention_mask(token_ids, valid_length)

        _, pooler = self.bert(input_ids = token_ids, token_type_ids = segment_ids.long(), attention_mask ₩
                              = attention_mask.float().to(token_ids.device))
        if self.dr_rate:
            out = self.dropout(pooler)
        else:
            out = pooler
        return self.classifier(out)
```

## *BERT 모델 + 학습 파라미터*

```
100% 62/62 [00:01<00:00, 54.25it/s]

epoch 9 val acc 0.859375

100% 246/246 [00:19<00:00, 12.93it/s]

epoch 10 batch id 1 loss 0.008023587986826897 train acc 1.0
epoch 10 batch id 201 loss 0.01610684022307396 train acc 0.9930814676616916
epoch 10 train acc 0.9932037601626016

100% 62/62 [00:01<00:00, 53.22it/s]

epoch 10 val acc 0.8601310483870968
```

**학습 데이터셋의 정확도는 거의 99%에 가까우며, 검증 데이터셋의 정확도는 86%인 것을 볼 수 있음**



***다음과 같은 문장을 입력으로 전달 받아 예측을 수행해보자***

```
comments = ['너무 우울해', '화가나서미쳐버리겠네', '아무것도하기싫다']

y_pred = getSentimentValue(comments, tok, max_len, batch_size, device)   # tok, max_len, batch_size, device
y_pred = list(map(int, y_pred))

print(intent_list[y_pred[0]], intent_list[y_pred[1]], intent_list[y_pred[2]])
```

```
우울감 분노 무기력
```

# 모델 성능 평가 - Confusion matrix

***Note***

## *다중 분류 문제의 경우, 혼동 행렬을 통해 모델의 성능 평가 진행*

| 실제 / 예측 | 우울감 | 슬픔 | 외로움 | 분노 | 무기력 | 감정조절이상 | 상실감 | 식욕저하 | 식욕증가 | 불면 | 초조함 | 피로 | 죄책감 | 집중력저하 | 자신감저하 | 자존감저하 | 절망감 | 자살충동 | 불안 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 우울감 | 285 | 4 | 5 | 1 | 13 | 0 | 0 | 3 | 1 | 9 | 3 | 0 | 1 | 1 | 2 | 1 | 0 | 0 | 2 |
| 슬픔 | 5 | 292 | 3 | 10 | 3 | 0 | 3 | 2 | 0 | 1 | 6 | 0 | 15 | 0 | 1 | 5 | 6 | 2 | 4 |
| 외로움 | 0 | 7 | 130 | 3 | 6 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 3 | 1 | 1 |
| 분노 | 0 | 9 | 1 | 154 | 2 | 6 | 1 | 0 | 0 | 2 | 10 | 0 | 11 | 0 | 3 | 7 | 0 | 8 | 3 |
| 무기력 | 9 | 9 | 2 | 1 | 202 | 1 | 0 | 1 | 0 | 2 | 5 | 8 | 2 | 0 | 3 | 2 | 6 | 5 | 0 |
| 감정조절이상 | 1 | 4 | 2 | 4 | 5 | 26 | 0 | 0 | 0 | 1 | 3 | 1 | 1 | 0 | 1 | 2 | 2 | 1 | 0 |
| 상실감 | 0 | 0 | 0 | 0 | 0 | 0 | 43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 식욕저하 | 1 | 0 | 0 | 0 | 4 | 0 | 0 | 204 | 2 | 2 | 4 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 식욕증가 | 0 | 1 | 0 | 0 | 2 | 1 | 0 | 7 | 71 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| 불면 | 2 | 0 | 2 | 0 | 4 | 0 | 0 | 2 | 0 | 341 | 7 | 10 | 0 | 0 | 0 | 1 | 1 | 1 | 2 |
| 초조함 | 4 | 3 | 1 | 6 | 3 | 4 | 0 | 1 | 1 | 3 | 473 | 3 | 2 | 0 | 3 | 4 | 3 | 4 | 22 |
| 피로 | 2 | 0 | 0 | 0 | 9 | 0 | 1 | 2 | 2 | 23 | 6 | 199 | 0 | 1 | 1 | 1 | 0 | 2 | 0 |
| 죄책감 | 1 | 11 | 2 | 2 | 5 | 2 | 1 | 0 | 0 | 0 | 7 | 2 | 191 | 0 | 4 | 4 | 1 | 1 | 2 |
| 집중력저하 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 1 | 40 | 1 | 0 | 0 | 0 | 2 |
| 자신감저하 | 0 | 1 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 1 | 1 | 49 | 3 | 1 | 0 | 1 |
| 자존감저하 | 0 | 1 | 0 | 1 | 3 | 1 | 0 | 0 | 0 | 1 | 5 | 0 | 2 | 0 | 10 | 219 | 1 | 1 | 0 |
| 절망감 | 4 | 0 | 3 | 2 | 3 | 1 | 0 | 0 | 0 | 0 | 3 | 0 | 1 | 0 | 3 | 4 | 113 | 6 | 1 |
| 자살충동 | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 121 | 0 |
| 불안 | 2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 161 |

```
f1 = round(f1_score(y_true, y_pred, average='micro'), 3)
f1
```

0.842

# *KoBERT -> KoELECTRA 모델 사용하기* *Note*

**KoELECTRA 모델을 Wellenss Dataset에 대해 사용하려면 여러 설정 값들을 바꿔주어야 함**

**KoELECTRA 모델의 nsmc (영화 리뷰 긍정 부정 예측 데이터셋) 모델을 19가지 우울증 분류 테스크로 바꿔보자**

## 1. 데이터셋 구조 변경

- nsmc 데이터셋은 다음과 같은 구조를 가짐

```
id      document        label
6270596 굳 ㅋ    1
9274899 GDNTOPCLASSINTHECLUB    0
8544678 뭐야 이 평점들은.... 나쁘진 않지만 10점 짜리는 더더욱 아니잖아      0
6825595 지루하지는 않은데 완전 막장임... 돈주고 보기에는....       0
6723715 3D만 아니었어도 별 다섯 개 줬을텐데.. 왜 3D로 나와서 제 심기를 불편하게 하죠??       0
7898805 음악이 주가 된, 최고의 음악영화    1
6315043 진정한 쓰레기    0
6097171 마치 미국애니에서 튀어나온듯한 창의력없는 로봇디자인부터가,고개를 젖게한다    0
8932678 갈수록 개판되가는 중국영화 유치하고 내용없음 폼잡다 끝남 말도안되는 무기에 유치한cg남무 아 그립다 동사서독같은 영화가 이건 3류아류작이다   0
6242223 이별의 아픔뒤에 찾아오는 새로운 인연의 기쁨 But, 모든 사람이 그렇지는 않네.. 1
7462111 괜찮네요오랜만포켓몬스터잼있어요   1
8425305 한국독립영화의 한계 그렇게 아버지가 된다와 비교됨   0
```

## 1. 데이터셋 구조 변경

- 웰니스 데이터셋도 다음과 같이 바꿔줌

X_train

| | id | context | label |
|---|---|---|---|
| **920** | 0000920 | 그래서 학원도 끊고 그냥 집에만 박혀 있는데 너무 우울해요. | 0 |
| **6152** | 0006152 | 모든 게 갑작스러워서 너무 힘든데 기댈 곳이 없어요. | 4 |
| **4674** | 0004674 | 남편 하는 행동 보면 갑자기 모든 게 허무하고 어이없는 것 같아 | 3 |
| **7052** | 0007052 | 내 주변에 사람이 너무 없는 거 같고 공허해 | 6 |
| **692** | 0000692 | 일도 알아보기 귀찮고 밤만 되면 우울해 | 0 |
| **...** | ... | ... | ... |
| **18700** | 0018700 | 한달동안 생각은 엄청 많이 했어요. 외국의 유명한 살인자들 보면 독극물살인도 많이 ... | 17 |
| **6872** | 0006872 | 나 너무 마음이 공허해 | 6 |
| **9757** | 0009757 | 자려고 누워도 계속 뒤척이기만 하고 잠을 못자서 찬송가 틀어놓고 자. | 9 |
| **18875** | 0018875 | 불안해서 잠도 안 와 | 18 |
| **14720** | 0014720 | 나 같은 사람이랑 일을 하다니 동료들이 불쌍해 | 12 |

15815 rows × 3 columns

```
array(['우울감', '슬픔', '외로움', '분노', '무기력', '감정조절이상', '상실감', '식욕저하', '식욕증가',
       '불면', '초조함', '피로', '죄책감', '집중력저하', '자신감저하', '자존감저하', '절망감', '자살충동',
       '불안'], dtype=object)
```

**label 0: 우울감, 1: 외로움, 2: 분노, ... , 18: 불안**

# KoBERT -> KoELECTRA 모델 사용하기

Note

KoELECTRA에서 사용하는 Config Class, Tokenizer Class, Sequence classifier model은 다음과 같음

```
from src import KoBertTokenizer, HanBertTokenizer
from transformers import (
    BertConfig,
    DistilBertConfig,
    ElectraConfig,
    XLMRobertaConfig,
    ElectraTokenizer,
    XLMRobertaTokenizer,
    BertForSequenceClassification,
    DistilBertForSequenceClassification,
    ElectraForSequenceClassification,
    XLMRobertaForSequenceClassification,
    BertForTokenClassification,
    DistilBertForTokenClassification,
    ElectraForTokenClassification,
    XLMRobertaForTokenClassification,
    BertForQuestionAnswering,
    DistilBertForQuestionAnswering,
    ElectraForQuestionAnswering,
    XLMRobertaForQuestionAnswering,
)
```

```
CONFIG_CLASSES = {
    "kobert": BertConfig,
    "distilkobert": DistilBertConfig,
    "hanbert": BertConfig,
    "koelectra-base": ElectraConfig,
    "koelectra-small": ElectraConfig,
    "koelectra-base-v2": ElectraConfig,
    "koelectra-base-v3": ElectraConfig,
    "koelectra-small-v2": ElectraConfig,
    "koelectra-small-v3": ElectraConfig,
    "xlm-roberta": XLMRobertaConfig,
}

TOKENIZER_CLASSES = {
    "kobert": KoBertTokenizer,
    "distilkobert": KoBertTokenizer,
    "hanbert": HanBertTokenizer,
    "koelectra-base": ElectraTokenizer,
    "koelectra-small": ElectraTokenizer,
    "koelectra-base-v2": ElectraTokenizer,
    "koelectra-base-v3": ElectraTokenizer,
    "koelectra-small-v2": ElectraTokenizer,
    "koelectra-small-v3": ElectraTokenizer,
    "xlm-roberta": XLMRobertaTokenizer,
}

MODEL_FOR_SEQUENCE_CLASSIFICATION = {
    "kobert": BertForSequenceClassification,
    "distilkobert": DistilBertForSequenceClassification,
    "hanbert": BertForSequenceClassification,
    "koelectra-base": ElectraForSequenceClassification,
    "koelectra-small": ElectraForSequenceClassification,
    "koelectra-base-v2": ElectraForSequenceClassification,
    "koelectra-base-v3": ElectraForSequenceClassification,
    "koelectra-small-v2": ElectraForSequenceClassification,
    "koelectra-small-v3": ElectraForSequenceClassification,
    "xlm-roberta": XLMRobertaForSequenceClassification,
}
```

## 2. 텍스트 분류에 사용되는 파이썬 파일 수정 - Wellness Dataset 프로세서 생성

- 웰니스 데이터셋에 대한 프로세서도 생성해줌 (seq_cls.py 파일 수정)

```python
class WellnessProcessor(object):
    """Processor for the Wellness data set """
    def __init__(self, args):
        self.args = args

    def get_labels(self):
        label_list = list(range(19))
        label_list = list(map(str, label_list))
        return label_list

    @classmethod
    def _read_file(cls, input_file):
        """Reads a comma separated value file (csv)."""
        with open(input_file, "r", encoding="utf-8") as f:
            lines = []
            for line in f:
                lines.append(line.strip(','))
            return lines

    def _create_examples(self, lines, set_type):
        """Creates examples for the training and dev sets."""
        examples = []
        for (i, line) in enumerate(lines[1:]):
            line = line.split("₩t")
            guid = "%s-%s" % (set_type, i)
            text_a = line[1]
            label = line[2]
            if i % 10000 == 0:
                logger.info(line)
            examples.append(InputExample(guid=guid, text_a=text_a, text_b=None, label=label))
        return examples

    def get_examples(self, mode):
        """
        Args:
            mode: train, dev, test
        """
```

```python
seq_cls_processors = {
    "kornli": KorNLIProcessor,
    "nsmc": NsmcProcessor,
    "paws": PawsProcessor,
    "korsts": KorSTSProcessor,
    "question-pair": QuestionPairProcessor,
    "hate-speech": HateSpeechProcessor,
    "Wellness": WellnessProcessor
}

seq_cls_tasks_num_labels = {"kornli": 3, "nsmc": 2, "paws": 2, "korsts": 1, "question-pair": 2, "hate-speech": 3, "Wellness": 19}

seq_cls_output_modes = {
    "kornli": "classification",
    "nsmc": "classification",
    "paws": "classification",
    "korsts": "regression",
    "question-pair": "classification",
    "hate-speech": "classification",
    "Wellness": "classfication"
}
```

## 3. KoELECTRA 환경 파일 수정

- koelectra 모델의 환경 파일을 수정해주자

```
{
  "task": "wellness",
  "data_dir": "data",
  "ckpt_dir": "ckpt",
  "train_file": "Wellness_Conversation_intent_train.tsv",
  "dev_file": "",
  "test_file": "Wellness_Conversation_intent_test.tsv",
  "evaluate_test_during_training": true,
  "eval_all_checkpoints": true,
  "save_optimizer": false,
  "do_lower_case": false,
  "do_train": true,
  "do_eval": true,
  "max_seq_len": 128,
  "num_train_epochs": 10,
  "weight_decay": 0.0,
  "gradient_accumulation_steps": 1,
  "adam_epsilon": 1e-8,
  "warmup_proportion": 0,
  "max_steps": -1,
  "max_grad_norm": 1.0,
  "no_cuda": false,
  "model_type": "koelectra-base",
  "model_name_or_path": "monologg/koelectra-base-discriminator",
  "output_dir": "koelectra-base-wellness-ckpt",
  "seed": 42,
  "train_batch_size": 32,
  "eval_batch_size": 128,
  "logging_steps": 2000,
  "save_steps": 2000,
  "learning_rate": 5e-5
}
```

## 4. 모델 학습 & 모델 정확도 평가

- 학습 횟수를 10으로 설정하고, KoELECTRA 모델을 학습해보자

```
08/05/2022 11:18:44 - INFO - processor.seq_cls -   Loading features from cached file data/cached_wellness_koelectra-base-discriminator_128
_train
08/05/2022 11:18:45 - INFO - processor.seq_cls -   Loading features from cached file data/cached_wellness_koelectra-base-discriminator_128
_test
/home/ubuntu/anaconda3/envs/chatbot/lib/python3.7/site-packages/transformers/optimization.py:310: FutureWarning: This implementation of Ad
amW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or set `no_deprecatio
n_warning=True` to disable this warning
  FutureWarning,
08/05/2022 11:18:46 - INFO - __main__ -   ***** Running training *****
08/05/2022 11:18:46 - INFO - __main__ -     Num examples = 15741
08/05/2022 11:18:46 - INFO - __main__ -     Num Epochs = 10
08/05/2022 11:18:46 - INFO - __main__ -     Total train batch size = 32
08/05/2022 11:18:46 - INFO - __main__ -     Gradient Accumulation steps = 1
08/05/2022 11:18:46 - INFO - __main__ -     Total optimization steps = 4920
08/05/2022 11:18:46 - INFO - __main__ -     Logging steps = 2000
08/05/2022 11:18:46 - INFO - __main__ -     Save steps = 2000
```

```
Epoch 1 done
Epoch 2 done
Epoch 3 done
Epoch 4 done
Epoch 5 done
Epoch 6 done
Epoch 7 done
Epoch 8 done
Epoch 9 done
Epoch 10 done
```

```
08/05/2022 11:21:12 - INFO - __main__ -   ***** Eval results on test dataset *****
08/05/2022 11:21:12 - INFO - __main__ -     acc = 0.8297764227642277
08/05/2022 11:21:12 - INFO - __main__ -   Saving model checkpoint to ckpt/koelectra-base-wellness-ckpt/checkpoint-2000
08/05/2022 11:23:35 - INFO - __main__ -   ***** Running evaluation on test dataset (4000 step) *****
08/05/2022 11:23:35 - INFO - __main__ -     Num examples = 3936
08/05/2022 11:23:35 - INFO - __main__ -     Eval Batch size = 128
```

```
100.00% [31/31 00:02<00:00]
```

```
08/05/2022 11:23:37 - INFO - __main__ -   ***** Eval results on test dataset *****
08/05/2022 11:23:37 - INFO - __main__ -     acc = 0.8488313008130082
08/05/2022 11:23:38 - INFO - __main__ -   Saving model checkpoint to ckpt/koelectra-base-wellness-ckpt/checkpoint-4000
08/05/2022 11:24:43 - INFO - __main__ -    global_step = 4920, average loss = 0.3510602872105665
```

**ckpt-2000을 이용한 모델 정확도: 0.83**
**ckpt-4000을 이용한 모델 정확도: 0.85**

**참고 사이트**
- https://github.com/monologg/KoELECTRA

# 기존 논문 챗봇 관련 구조

Note

## 감성 분류 학습용 데이터

AI Hub 감성대화 말뭉치
- 상황(주제) – 감정 대분류 – 감정 분류 – 사람 문장 – 답변으로 구성
- 상황(주제) 카테고리 12개
- 감정대분류: 기쁨, 당황, 분노, 불안, 상처,슬픔 으로 구성

챗봇 답변 없음

AI Hub 한국어 감정 정보가 포함된 단발성 대화
- 질의 – 감정으로 구성
- 감정: 기쁨, 슬픔, 놀람, 분노, 공포, 혐오, 중립 (7가지)

## 주제 분류/챗봇 학습 데이터

AI Hub 웰니스 상담 데이터
- 주제 – 질문 – 답변으로 구성
- 주제 카테고리 (359개)
- 감성 label 없음
- 공감형 답변

Songys 이별관련 챗봇 데이터
- 질문 – 답변 – 감성으로 구성
- 감성 레이블 일치 필요
- 주제 label 없음
- 공감형 답변

우울증 질의응답
- 질문 – 답변으로 구성
- 감성 label 없음
- 주제 label 없음
- 공감형 답변보다는 솔루션 제공

AI Hub 감성대화 말뭉치
- 상황(주제) – 감정 대분류 – 감정 분류 – 사람 문장 – 답변으로 구성
- 상황(주제) 카테고리 12개
- 감정대분류: 기쁨, 당황, 분노, 불안, 상처,슬픔 으로 구성
- 공감형 답변

## Sentiment labeling

word2vec

AI Hub 감성대화 말뭉치 & AI Hub 한국어 감정정보 단발성 대화
- 부정 감성 질문만 도출
- Word2vec으로 상단 우울 감성 레이블과, 출현한 단어들 사이의 vector similarity 계산

Topic labeling

감성 레이블링이 필요한 데이터셋
- AI Hub 웰니스 상담 데이터
- Songys 이별관련 챗봇 데이터
- 우울증 질의응답
- AI Hub 감성대화 말뭉치

**우울 감성 레이블 ( 미정 )**
: 우울, 힘듦, 불안,무서움, 걱정, 답답함, 짜증, 외로움, 괴로움, 슬픔, 속상함, 화남, 무기력, 혼란스러움

**주제 레이블**
: AI Hub 웰니스 상담 데이터에서 사용된 **359개의 주제 카테고리로 통일**

## Topic labeling

웰니스 상담데이터 359개 주제

LDA Topic modeling → word2vec → Topic labeling

AI Hub 웰니스 상담 데이터
- 이미 구분지어진 359개 카테고리별로 질문을 묶어 topic modeling 진행
- 토픽모델링 결과로 나온 359개 토픽의 대표 단어들을 도출 (예. 5개씩)
- Word2vec으로 해당 5개의 단어들과, 출현한 단어들 사이의 vector similarity 계산

주제 레이블링이 필요한 데이터셋
- Songys 이별 관련 챗봇 데이터
- 우울증 질의응답
- AI Hub 감성대화 말뭉치

**감성 분류 학습 데이터 & 주제 분류 학습 데이터**
- AI Hub 웰니스 상담 데이터
- Songys 이별관련 챗봇 데이터
- AI Hub 감성대화 말뭉치

**챗봇 학습 데이터**
- AI Hub 웰니스 상담 데이터
- Songys 이별관련 챗봇 데이터
- 우울증 질의응답
- AI Hub 감성대화 말뭉치

**챗봇 성능 평가 데이터**
- AI Hub 한국어 감정 정보가 포함된 단발성 대화

*앞선 논문 구조를 참고해, 토픽 모델링 진행하여 성능 비교해볼 것*