```
!pip install transformers[torch]
!git clone https://github.com/ZIZUN/korean-malicious-comments-dataset.git
import pandas as pd
import matplotlib.pyplot as plt

import torch
from transformers import AutoTokenizer, AutoModelForSequenceClassification, Trair
from sklearn.metrics import precision_recall_fscore_support, accuracy_score

device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
print("device:", device)
```

```
Requirement already satisfied: transformers[torch] in /usr/local/lib/python3.
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: huggingface-hub<1.0,>=0.16.4 in /usr/local/lib
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/d
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: tokenizers<0.19,>=0.14 in /usr/local/lib/pytho
Requirement already satisfied: safetensors>=0.3.1 in /usr/local/lib/python3.1
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: torch!=1.12.0,>=1.10 in /usr/local/lib/python3
Requirement already satisfied: accelerate>=0.20.3 in /usr/local/lib/python3.1
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/p
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: triton==2.1.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/pyt
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.1
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.1
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/d
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.10/dist
fatal: destination path 'korean-malicious-comments-dataset' already exists an
device: cuda:0
```

```
df = pd.read_csv("/content/korean-malicious-comments-dataset/Dataset.csv", sep="\
df.head()
```

	content	lable
0	이종석 한효주 나오는 드라마 이후로 드라마 안봤다. 2년전인가?? 좀 신선했었지. ...	0.0
1	씨바알..노무노무 술프노... 오늘 저녁은 꽃등심이다ㅠㅜ	0.0
2	짱깨 꺼라_패쓰	0.0
3	그들의 사생활 ~ 고인이된 설리를 위해서라도 모두 조용하길 지금 누굴 탓한다고 무슨...	1.0
4	아무리 법이 뭣같아도 무슨 자격으로 개인의 신상정보를 불특정 다수에게 공개하는지 도...	1.0

Next steps: Generate code with df View recommended plots

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   content  10000 non-null  object 
 1   lable    9975 non-null   float64
dtypes: float64(1), object(1)
memory usage: 156.4+ KB
```

```
null_idx = df[df.lable.isnull()].index
df.loc[null_idx, "content"]
```

```
1602    응애 응애 엄마 저 맘에 안들죠? .....아들 ?? " 너 내가 우스워 보이...
1654            토니스타크 평소 "아이엠그루트"라는 유행어를 부러워했다는게 학계의 정설\t1
1992    "13일 현대차에 따르면 올 들어 국내 소비자들의 수입차 구매의향률이 3년 만에 하...
2920                  에이프릴이 한마디 합니다 "예쁜게 죄" 구하라님 "무기징역"\t1
3720          답글 글씨체를 봐라 저게 애새끼가 쓴거냐?"빨갱이새끼가 쓴거지 ㅁㅈㅎㅉㅉ\t0
3807    알겠다이기ㅋㅋ 딱 채찍쳐맞는거 좋아하는 한국식 마인드네. 노예마인드. 조금만 성공한...
3908            이래서 스스로 걸리거든 "죄인들이"~ㅎㅎㅎ 재미보고 털리고 그치~~~?\t0
4241    아버지는 내재된 악마들을 다룰 정신적 힘을 가지고 있지 않았다." 이 말한마디가 사...
4283    댓글 중 "선동 당해서 촛불든 개돼지 홍어들도 단죄를 받아야 할 공범자들이다"에10...
5000    스파이 제안받고 살해 안당하는 법1. 처음에 스파이 제안을 받았을때 "중국을 위해서...
5521    "국방부 "까지 _ㄱ ㅐ 엿같은 ㅈ ㅣ랄주댕이...좌빨에서 _인민군대로 _가려는건가...
5866    쌩뚱맞게 60대최반엌 치매라니 그것도 곱게 사는 사모님이- -" 알콜중독도 아니고 ...
6477    페미메퇘지쿵쾅년인 메갈페미들은 니들이 좋아하는 싫어요 ㄱㄱ제발부탁해~~"일반 여성"...
6538    아니 ㅆㅂ 그런 "카더라"가 넘쳐난다고 그거에 대해서 혹시 댓글게이는 뭔가 아는거 ...
6771    저 때 투니버스에서 코요태 짧게 인터뷰 했었는데 김종민이 "노래는 뭐 신지가 다 하...
6932                개 족 가튼 국방부의 "휴기연장콜센터"발족을 축하한다 ㅆ ㅂ..\t0
7199    민족적 자존심과 애국심을 갖고 국산품 이용합시다 . . . "겸손"한 마음으로 재산...
7252    아나운서는 목표가 아니었지ㅋㅋ재벌하고 결혼하자마자 바로 은퇴하네ㅋㅋ무슨 인터뷰한 거...
7270    결국 준영과 다솜은 바람을 피게되고 무인도로 떠난다에 한표 ㅋㅋㅋ 자연인이 되어 "...
7480    지금 연락하는 여자랑 폰섹 엄청 많이했는데만나서 호텔 들어가서침대에 서로 마주보고 ...
7499    몽골한테 "최근에" 250년간 지배당하고 집단강간을 당했는데 동양피가 하나도 안섞였...
7887    뭐 선천적으로 여성스럽거나 여자역할을 하고 싶어하는 동성애자들 그럴 수 있다고는 생...
9666          ㄹㅇ 시발 그냥 "다른 진로 생각해 보세요"라고만 했어도 욕 안처 먹었지.\t0
9698                              간만에 이단어가 떠오르는군 "이뭐병"\t0
9875    노라조 "형"이란 노래로 힘들 때 위로를 받곤 했습니다. 앞으로도 노라조라는 이름으...
Name: content, dtype: object
```

```
df.loc[null_idx, "lable"] = df.loc[null_idx, "content"].apply(lambda x: x[-1])
df.loc[null_idx, "content"] = df.loc[null_idx, "content"].apply(lambda x: x[:-2])
```

```
df = df.astype({"lable":"int"})
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   content  10000 non-null  object
 1   lable    10000 non-null  int64 
dtypes: int64(1), object(1)
memory usage: 156.4+ KB
```

데이터셋 준비

```
train_data = df.sample(frac=0.8, random_state=42)
test_data = df.drop(train_data.index)
```

```
# 데이터셋 갯수 확인
print('중복 제거 전 학습 데이터셋 : {}'.format(len(train_data)))
print('중복 제거 전 테스트 데이터셋 : {}'.format(len(test_data)))

# 중복 데이터 제거
train_data.drop_duplicates(subset=["content"], inplace= True)
test_data.drop_duplicates(subset=["content"], inplace= True)

# 데이터셋 갯수 확인
print('중복 제거 후 학습 데이터셋 : {}'.format(len(train_data)))
print('중복 제거 후 테스트 데이터셋 : {}'.format(len(test_data)))
```

```
중복 제거 전 학습 데이터셋 : 8000
중복 제거 전 테스트 데이터셋 : 2000
중복 제거 후 학습 데이터셋 : 7992
중복 제거 후 테스트 데이터셋 : 2000
```

```
MODEL_NAME = "beomi/KcELECTRA-base"
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: U
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings ta
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access p
  warnings.warn(
```

```
tokenized_train_sentences = tokenizer(
    list(train_data["content"]),
    return_tensors="pt",
    max_length=128,
    padding=True,
    truncation=True,
    add_special_tokens=True,
    )
```

```
print(tokenized_train_sentences[0])
print(tokenized_train_sentences[0].tokens)
print(tokenized_train_sentences[0].ids)
print(tokenized_train_sentences[0].attention_mask)
```

```
Encoding(num_tokens=128, attributes=[ids, type_ids, tokens, offsets, attentio
['ĠêµŃë°©ë¶Ģ', '~~', 'ìłĦ', 'íĻĶëıĽ', 'ĠíĽ´ê°Ģ', 'ìĹ°', 'ìłľìĿĦ', 'ĠíķĺĽ', 'Ġë
[7727, 424, 497, 9927, 6413, 712, 3101, 372, 1566, 5898, 16745, 9129, 11079,
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0
```

```
tokenized_test_sentences = tokenizer(
    list(test_data["content"]),
    return_tensors="pt",
    max_length=128,
    padding=True,
    truncation=True,
    add_special_tokens=True,
    )
```

```
class CurseDataset(torch.utils.data.Dataset):
    def __init__(self, encodings, labels):
        self.encodings = encodings
        self.labels = labels

    def __getitem__(self, idx):
        item = {key: torch.tensor(val[idx]) for key, val in self.encodings.items(
        item["labels"] = torch.tensor(self.labels[idx])
        return item

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

```
train_label = train_data["lable"].values
test_label = test_data["lable"].values

train_dataset = CurseDataset(tokenized_train_sentences, train_label)
test_dataset = CurseDataset(tokenized_test_sentences, test_label)
```

학습

```
model = AutoModelForSequenceClassification.from_pretrained(MODEL_NAME, num_labels
model.to(device)
```

```
Some weights of ElectraForSequenceClassification were not initialized from th
You should probably TRAIN this model on a down-stream task to be able to use
ElectraForSequenceClassification(
  (electra): ElectraModel(
    (embeddings): ElectraEmbeddings(
      (word_embeddings): Embedding(30000, 768, padding_idx=3)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): ElectraEncoder(
      (layer): ModuleList(
        (0-11): 12 x ElectraLayer(
          (attention): ElectraAttention(
            (self): ElectraSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): ElectraSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12,
elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): ElectraIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): ElectraOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12,
elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
    )
  )
  (classifier): ElectraClassificationHead(
    (dense): Linear(in_features=768, out_features=768, bias=True)
    (dropout): Dropout(p=0.1, inplace=False)
    (out_proj): Linear(in_features=768, out_features=2, bias=True)
  )
)
```

```
training_args = TrainingArguments(
    output_dir='./',
    num_train_epochs=10,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=64,
    logging_dir='./logs',
    logging_steps=500,
    save_total_limit=2,
)
```

```
def compute_metrics(pred):
    labels = pred.label_ids
    preds = pred.predictions.argmax(-1)
    precision, recall, f1, _ = precision_recall_fscore_support(labels, preds, ave
    acc = accuracy_score(labels, preds)
    return {
        'accuracy': acc,
        'f1': f1,
        'precision': precision,
        'recall': recall
    }
```

```
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=test_dataset,
    compute_metrics=compute_metrics,
)
```

```
trainer.train()
```

```
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
```

[9990/9990 36:42, Epoch 10/10]

Step	Training Loss
500	0.417600
1000	0.332200
1500	0.236400
2000	0.237400
2500	0.175400
3000	0.168200
3500	0.128000
4000	0.136200
4500	0.099500
5000	0.089700
5500	0.073900
6000	0.076600
6500	0.057600
7000	0.070400
7500	0.038200
8000	0.043200
8500	0.024800
9000	0.041800
9500	0.033100

```
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
```

```
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.item
TrainOutput(global_step=9990, training_loss=0.1253599260423754, metrics=
{'train_runtime': 2206.0514, 'train_samples_per_second': 36.228,
'train_steps_per_second': 4.528, 'total_flos': 5256958886092800.0,
'train_loss': 0.1253599260423754, 'epoch': 10.0})
```

평가

```
trainer.evaluate(eval_dataset=test_dataset)
```

```
<ipython-input-14-b5e02a6514c5>:7: UserWarning: To copy construct from a tens
  item = {key: torch.tensor(val[idx]) for key, val in self.encodings.items()}
[32/32 00:14]
{'eval_loss': 0.5940119624137878,
 'eval_accuracy': 0.9055,
 'eval_f1': 0.904977375565611,
 'eval_precision': 0.8973080757726819,
 'eval_recall': 0.9127789046653144,
 'eval_runtime': 15.1354,
 'eval_samples_per_second': 132.141,
 'eval_steps_per_second': 2.114,
 'epoch': 10.0}
```

```
# 0: curse, 1: non_curse
def sentence_predict(sent):
    model.eval()

    tokenized_sent = tokenizer(
        sent,
        return_tensors="pt",
        truncation=True,
        add_special_tokens=True,
        max_length=128
    )

    tokenized_sent.to(device)
```