🏆

1122 수상작 리뷰

주제

Your goal is to predict whether a given hypothesis is related to its premise by `contradiction`, `entailment`, or whether neither of those is true (`neutral`).

https://www.kaggle.com/competitions/contradictory-my-dear-watson/overview

데이터

- **train.csv**: ID, premise, hypothesis, and label, language of the text and its two-letter abbreviation
- **test.csv**: ID, premise, hypothesis, language, language abbreviation without label
- **sample_submission.csv**: ID, label

코드 정리

전처리

```
import numpy as np
import pandas as pd
from tensorflow import keras
import keras_nlp
import seaborn as sns
import matplotlib.pyplot as plt
import os

print("TensorFlow version:", tf.__version__)
print("KerasNLP version:", keras_nlp.__version__)
```

```
from transformers import AutoModelForSequenceClassification, Tra
model = AutoModelForSequenceClassification.from_pretrained("xlm-
```

```
train_df=pd.read_csv("../input/contradictory-my-dear-watson/trai
train_df=train_df.sample(frac=1.0)
val_df=train_df.iloc[:2000, :]
train_df=train_df.iloc[2000:, :]
train_df['input_text']=train_df['premise']+' [SEP] '+train_df['h
val_df['input_text']=val_df['premise']+' [SEP] '+val_df['hypothe
```

```
from datasets import Dataset
train_ds=Dataset.from_pandas(train_df)
val_ds=Dataset.from_pandas(val_df)

train_ds
```

모델링

```
from sklearn.metrics import accuracy_score, f1_score
def compute_metrics(pred):
    labels=pred.label_ids
    preds=pred.predictions.argmax(-1)
    f1=f1_score(labels, preds, average='weighted')
    ac=accuracy_score(labels, preds)
    return {"accuracy":ac, "f1":f1}
```

```
os.environ["WANDB_DISABLED"] = "true"
```

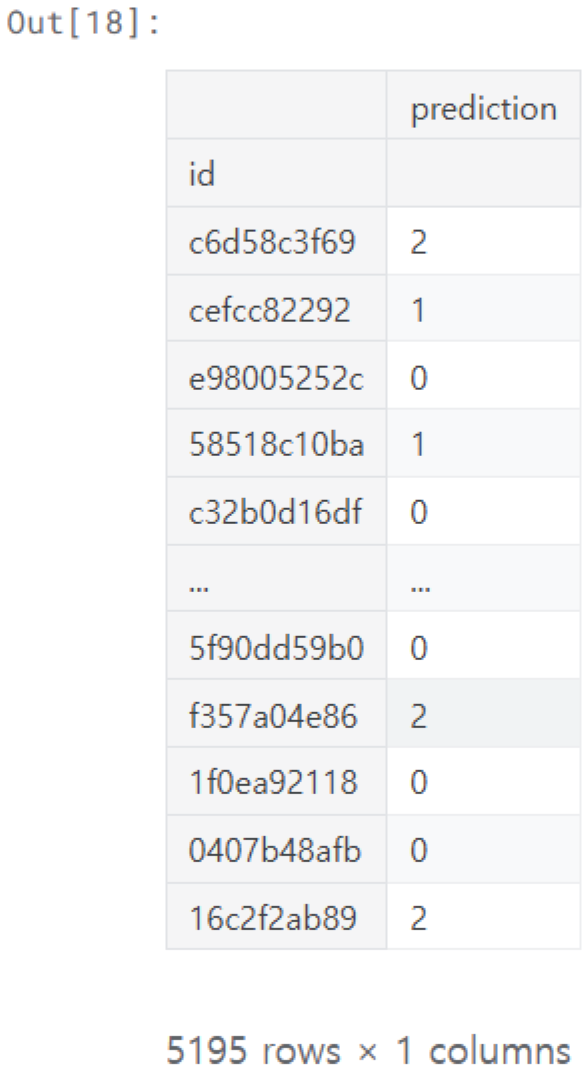
```
training_args = TrainingArguments(
    output_dir="./results",
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=1,
    weight_decay=0.01,
)

trainer = Trainer(
    model=model,
    args=training_args,
    compute_metrics=compute_metrics,
    train_dataset=train_ds_encoded,
    eval_dataset=val_ds_encoded,
    tokenizer=tokenizer,
    data_collator=data_collator,
)

trainer.train()
```

제출 파일에 저장

```
test_df=pd.read_csv("../input/contradictory-my-dear-watson/test
test_df['input_text']=test_df['premise']+' [SEP] '+test_df['hypo
test_ds=Dataset.from_pandas(test_df)
test_ds_encoded=test_ds.map(preprocess_function, batched=True)
```

```
test_output=trainer.predict(test_ds_encoded)
outputs=test_output.predictions.argmax(-1)
sub_df=pd.DataFrame({"id": test_df["id"], "prediction": outpu
ts})
sub_df.set_index("id", inplace=True)
sub_df.to_csv("submission.csv")
```

```
sub_df
```



Out[18]:

	prediction
id	
c6d58c3f69	2
cefcc82292	1
e98005252c	0
58518c10ba	1
c32b0d16df	0
...	...
5f90dd59b0	0
f357a04e86	2
1f0ea92118	0
0407b48afb	0
16c2f2ab89	2

5195 rows × 1 columns

차별점 및 배울점

- ***os.environ["WANDB_DISABLED"] = "true"*** 를 통해 Weights&Biases(W&B)라는 머신러닝 실험 추적 도구를 비활성화한 부분이 인상적이었다. 이 과정을 통해 로그 전송 및 클라우드 통신을 비활성화할 수 있다.

- 이 코드에서는 **Hugging Face Transformers 라이브러리**를 사용하였는데, 이 라이브러리는 딥레닝 모델의 학습 및 평가를 매우 유연하고 편리하게 만들어주었다. 특히 weight_decay=0.01로 설정하여 모델의 가중치를 줄여 일반화를 향상시켰다.