Name : Devdeep Shetranjiwala
Email ID : devdeep0702@gmail.com

Screening exercise

> **When submitting your application please also complete the following exercise. Write a Jupyter Notebook to conducting a small task with a transformer and explain what you are trying to solve.**

(Please check the installation, examples, and tutorial if needed: https://huggingface.co/docs/transformers/index)

The goal of this task was to classify sentences as either grammatically correct or incorrect using a pre-trained transformer model from the Hugging Face Transformers library, fine-tuned on the CoLA dataset. The CoLA dataset is a corpus of English sentences labeled with a binary acceptability judgment indicating whether the sentence is grammatically correct or incorrect. The task involves natural language processing (NLP) and binary classification.

We used the BERT (Bidirectional Encoder Representations from Transformers) model, which is a pre-trained transformer model that has achieved state-of-the-art results on a wide range of NLP tasks. We fine-tuned the pre-trained BERT model on the CoLA dataset using TensorFlow, which is an open-source platform for machine learning that provides a high-level API for building and training machine learning models.

By fine-tuning the pre-trained BERT model on the CoLA dataset, we were able to leverage the pre-trained model's knowledge of natural language to improve the accuracy of our classification task. We evaluated the performance of the model on the validation dataset and used it to make predictions on new sentences.

The ability to accurately classify sentences as grammatically correct or incorrect has many practical applications in NLP, such as in automated essay grading, grammar checking, and language translation.

In [ ]:

```
!pip install tensorflow
!pip install transformers

import tensorflow as tf
from transformers import TFBertForSequenceClassification, BertTokenizer

# Download the dataset
!wget https://nyu-mll.github.io/CoLA/cola_public_1.1.zip
!unzip cola_public_1.1.zip
# Next, we will load the dataset and preprocess it:

import pandas as pd

# Load the dataset
train_df = pd.read_csv("cola_public/tokenized/in_domain_train.tsv", delimiter="\t", header=None, names=["sentence_source", "label", "label_notes", "sentence"])

# Preprocess the dataset
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

def preprocess_data(data):
    sentences = data["sentence"].tolist()
    labels = data["label"].tolist()
    labels = [0 if label == 0 else 1 for label in labels]   # Convert label 2 to label 1
    encodings = tokenizer(sentences, truncation=True, padding=True)
    return tf.data.Dataset.from_tensor_slices((dict(encodings), labels))

train_data = preprocess_data(train_df)
# Now, we will fine-tune the pre-trained BERT model on the CoLA dataset:

# Create the model
model = TFBertForSequenceClassification.from_pretrained("bert-base-uncased", num_labels=2
)

# Define the optimizer and loss function
optimizer = tf.keras.optimizers.Adam(learning_rate=5e-5)
loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)

# Compile the model
model.compile(optimizer=optimizer, loss=loss, metrics=["accuracy"])

# Train the model
model.fit(train_data.shuffle(1000).batch(16), epochs=3)

# Fine-tuned a pre-trained BERT model on the CoLA dataset using TensorFlow and Hugging Fa
ce Transformers.
# The trained model can now be used to classify new sentences as either grammatically cor
rect or incorrect.
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/publi
c/simple/
Requirement already satisfied: tensorflow in /usr/local/lib/python3.9/dist-packages (2.11
.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.9/dist-package
s (from tensorflow) (1.6.3)
Requirement already satisfied: tensorflow-estimator<2.12,>=2.11.0 in /usr/local/lib/pytho
n3.9/dist-packages (from tensorflow) (2.11.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.9/dist-packages
(from tensorflow) (2.2.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.9/dist-
packages (from tensorflow) (4.5.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.9/dist-packages (f
rom tensorflow) (1.15.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.9/dist-packages (fro
m tensorflow) (3.8.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.9/dist-packages (from
tensorflow) (23.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.9/dist-packages (fro
m tensorflow) (1.16.0)
Requirement already satisfied: keras<2.12,>=2.11.0 in /usr/local/lib/python3.9/dist-packa
ges (from tensorflow) (2.11.0)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.9/dist-packages (fro
m tensorflow) (1.22.4)
Requirement already satisfied: setuptools in /usr/local/lib/python3.9/dist-packages (from
tensorflow) (67.6.1)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.9/dist-packa
ges (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.9/dist-packages
(from tensorflow) (16.0.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.9/dist-packa
ges (from tensorflow) (1.51.3)
Requirement already satisfied: tensorboard<2.12,>=2.11 in /usr/local/lib/python3.9/dist-p
ackages (from tensorflow) (2.11.2)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.9/dist-packa
ges (from tensorflow) (0.4.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.9/dist-packages (
from tensorflow) (1.4.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/pyt
hon3.9/dist-packages (from tensorflow) (0.31.0)
Requirement already satisfied: flatbuffers>=2.0 in /usr/local/lib/python3.9/dist-packages
(from tensorflow) (23.3.3)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in /usr/local/lib/python3.9/dist-pac
kages (from tensorflow) (3.19.6)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.9/dist-package
s (from tensorflow) (3.3.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.9/dist-packag
es (from astunparse>=1.6.0->tensorflow) (0.40.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.9/
dist-packages (from tensorboard<2.12,>=2.11->tensorflow) (1.8.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.9/dist-packages
(from tensorboard<2.12,>=2.11->tensorflow) (3.4.3)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.9/dist-packa
ges (from tensorboard<2.12,>=2.11->tensorflow) (2.27.1)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3
```

```
.9/dist-packages (from tensorboard<2.12,>=2.11->tensorflow) (0.4.6)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.9/dist-packages
(from tensorboard<2.12,>=2.11->tensorflow) (2.2.3)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.9/dist-pac
kages (from tensorboard<2.12,>=2.11->tensorflow) (2.16.3)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/py
thon3.9/dist-packages (from tensorboard<2.12,>=2.11->tensorflow) (0.6.1)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.9/dist-pa
ckages (from google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (5.3.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.9/dist-pac
kages (from google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.9/dist-packages (f
rom google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.9/dist-
packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11->tensorflow) (1.
3.1)
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.9/dist-p
ackages (from markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow) (6.1.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist
-packages (from requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (fr
om requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (3.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packag
es (from requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-pac
kages (from requests<3,>=2.21.0->tensorboard<2.12,>=2.11->tensorflow) (1.26.15)
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.9/dist-package
s (from werkzeug>=1.0.1->tensorboard<2.12,>=2.11->tensorflow) (2.1.2)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.9/dist-packages (from
importlib-metadata>=4.4->markdown>=2.6.8->tensorboard<2.12,>=2.11->tensorflow) (3.15.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.9/dist-pack
ages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.12,>=2.11->tensorf
low) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.9/dist-packages
(from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard<2.12,>=2.11
->tensorflow) (3.2.2)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/publi
c/simple/
Requirement already satisfied: transformers in /usr/local/lib/python3.9/dist-packages (4.
27.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.9/dist-packages (fro
m transformers) (1.22.4)
Requirement already satisfied: filelock in /usr/local/lib/python3.9/dist-packages (from t
ransformers) (3.10.7)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.9/dist-packages (fro
m transformers) (6.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-packages
(from transformers) (23.0)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in /usr/local/lib/python
3.9/dist-packages (from transformers) (0.13.2)
Requirement already satisfied: requests in /usr/local/lib/python3.9/dist-packages (from t
ransformers) (2.27.1)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.9/dist-packages (from
transformers) (4.65.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.11.0 in /usr/local/lib/python3.9/d
ist-packages (from transformers) (0.13.3)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.9/dist-package
s (from transformers) (2022.10.31)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.9/dis
t-packages (from huggingface-hub<1.0,>=0.11.0->transformers) (4.5.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.9/dist-pac
kages (from requests->transformers) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.9/dist-packag
es (from requests->transformers) (2022.12.7)
Requirement already satisfied: charset-normalizer~=2.0.0 in /usr/local/lib/python3.9/dist
-packages (from requests->transformers) (2.0.12)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.9/dist-packages (fr
om requests->transformers) (3.4)
--2023-03-30 06:32:50--  https://nyu-mll.github.io/CoLA/cola_public_1.1.zip
Resolving nyu-mll.github.io (nyu-mll.github.io)... 185.199.108.153, 185.199.109.153, 185.
199.110.153, ...
Connecting to nyu-mll.github.io (nyu-mll.github.io)|185.199.108.153|:443... connected.
```

```
HTTP request sent, awaiting response... 200 OK
Length: 255330 (249K) [application/zip]
Saving to: ‘cola_public_1.1.zip.1’

cola_public_1.1.zip 100%[===================>] 249.35K  --.-KB/s    in 0.004s

2023-03-30 06:32:50 (60.7 MB/s) - ‘cola_public_1.1.zip.1’ saved [255330/255330]

Archive:  cola_public_1.1.zip
replace cola_public/README? [y]es, [n]o, [A]ll, [N]one, [r]ename: N
```

```
All model checkpoint layers were used when initializing TFBertForSequenceClassification.

Some layers of TFBertForSequenceClassification were not initialized from the model checkp
oint at bert-base-uncased and are newly initialized: ['classifier']
You should probably TRAIN this model on a down-stream task to be able to use it for predi
ctions and inference.
```

```
Epoch 1/3
535/535 [==============================] - 3068s 6s/step - loss: 0.5292 - accuracy: 0.737
9
Epoch 2/3
535/535 [==============================] - 3013s 6s/step - loss: 0.3441 - accuracy: 0.857
8
Epoch 3/3
535/535 [==============================] - 2998s 6s/step - loss: 0.2047 - accuracy: 0.925
2
```

Out[ ]:

```
<keras.callbacks.History at 0x7f79807b4e50>
```

In [ ]:

```
# Load the validation dataset
val_df = pd.read_csv("cola_public/tokenized/in_domain_dev.tsv", delimiter="\t", header=N
one, names=["sentence_source", "label", "label_notes", "sentence"])

# Preprocess the validation dataset
val_data = preprocess_data(val_df)

# Evaluate the model on the validation dataset
model.evaluate(val_data.batch(16))
```

```
33/33 [==============================] - 39s 1s/step - loss: 0.5099 - accuracy: 0.8254
```

Out[ ]:

```
[0.5098571181297302, 0.8254269361495972]
```

> **we can do some testing for this model by evaluating it on the CoLA validation dataset. Here's how we can do that This will output the model's loss and accuracy on the validation dataset. We can also use the model to make predictions on new sentences**

In [ ]:

```
# Example sentence
sentence = "The cat is sleeping on the mat."

# Preprocess the sentence
input_ids = tokenizer.encode(sentence, return_tensors="tf")
input_dict = {"input_ids": input_ids, "attention_mask": tf.ones_like(input_ids)}

# Make a prediction
prediction = tf.nn.softmax(model(input_dict)[0], axis=1)

# Print the predicted label and probability distribution
```

```python
labels = ["grammatically incorrect", "grammatically correct"]
print(f"Sentence: {sentence}")
print(f"Predicted label: {labels[prediction.numpy().argmax()]}")
print(f"Probability distribution: {prediction.numpy()[0]}")
```

```
Sentence: The cat is sleeping on the mat.
Predicted label: grammatically correct
Probability distribution: [0.00481604 0.995184  ]
```

In [ ]:

```python
# Example sentence
sentence = "Me is Devdeep."

# Preprocess the sentence
input_ids = tokenizer.encode(sentence, return_tensors="tf")
input_dict = {"input_ids": input_ids, "attention_mask": tf.ones_like(input_ids)}

# Make a prediction
prediction = tf.nn.softmax(model(input_dict)[0], axis=1)

# Print the predicted label and probability distribution
labels = ["grammatically incorrect", "grammatically correct"]
print(f"Sentence: {sentence}")
print(f"Predicted label: {labels[prediction.numpy().argmax()]}")
print(f"Probability distribution: {prediction.numpy()[0]}")
```

```
Sentence: Me is Devdeep.
Predicted label: grammatically incorrect
Probability distribution: [0.9743079  0.02569204]
```