Bert_Model_IMDB

April 27, 2021

1 Computational Intelligence Project: Sentiment Analysis on IMDB dataset Using BERT Model

Sentiment analysis is one of the key areas of research in NLP and Sequence modelling. I will be using fuzzy systems to predict two classes - positive or negative sentiment.

[ ]:
```
!pip install transformers
```

```
Collecting transformers
  Downloading https://files.pythonhosted.org/packages/ed/d5/f4157a376b8a79
489a76ce6cfe147f4f3be1e029b7144fa7b8432e8acb26/transformers-4.4.2-py3-none-
any.whl (2.0MB)
     || 2.0MB 8.7MB/s
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-
packages (from transformers) (2.23.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.7
/dist-packages (from transformers) (2019.12.20)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in
/usr/local/lib/python3.7/dist-packages (from transformers) (3.8.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-
packages (from transformers) (20.9)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-
packages (from transformers) (1.19.5)
Collecting sacremoses
  Downloading https://files.pythonhosted.org/packages/7d/34/09d19aff26edcc
8eb2a01bed8e98f13a1537005d31e95233fd48216eed10/sacremoses-0.0.43.tar.gz (883kB)
     || 890kB 47.8MB/s
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.7
/dist-packages (from transformers) (4.41.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-
packages (from transformers) (3.0.12)
Collecting tokenizers<0.11,>=0.10.1
  Downloading https://files.pythonhosted.org/packages/71/23/2ddc317b212111
7bf34dd00f5b0de194158f2a44ee2bf5e47c7166878a97/tokenizers-0.10.1-cp37-cp37m-
manylinux2010_x86_64.whl (3.2MB)
     || 3.2MB 52.8MB/s
Requirement already satisfied: certifi>=2017.4.17 in
```

```
/usr/local/lib/python3.7/dist-packages (from requests->transformers) (2020.12.5)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.7/dist-packages (from requests->transformers) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-
packages (from requests->transformers) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7
/dist-packages (from requests->transformers) (3.0.4)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-
packages (from importlib-metadata; python_version < "3.8"->transformers) (3.4.1)
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.8"
in /usr/local/lib/python3.7/dist-packages (from importlib-metadata;
python_version < "3.8"->transformers) (3.7.4.3)
Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.7
/dist-packages (from packaging->transformers) (2.4.7)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.15.0)
Requirement already satisfied: click in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (7.1.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages
(from sacremoses->transformers) (1.0.1)
Building wheels for collected packages: sacremoses
  Building wheel for sacremoses (setup.py) ... done
  Created wheel for sacremoses: filename=sacremoses-0.0.43-cp37-none-any.whl
size=893262
sha256=d5538b49b30c8b49459bde73fb51edd5baec911261d2df8e7577f1a6391a0699
  Stored in directory: /root/.cache/pip/wheels/29/3c/fd/7ce5c3f0666dab31a5012363
5e6fb5e19ceb42ce38d4e58f45
Successfully built sacremoses
Installing collected packages: sacremoses, tokenizers, transformers
Successfully installed sacremoses-0.0.43 tokenizers-0.10.1 transformers-4.4.2
```

1.1 Loading BERT Model

[ ]:
```
from transformers import BertTokenizer, TFBertForSequenceClassification
from transformers import InputExample, InputFeatures

model = TFBertForSequenceClassification.from_pretrained("bert-base-uncased")
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")
```

```
HBox(children=(FloatProgress(value=0.0, description='Downloading', max=433.0, style=ProgressSty
```

```
HBox(children=(FloatProgress(value=0.0, description='Downloading', max=536063208.0, style=Progr
```

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

Some layers of TFBertForSequenceClassification were not initialized from the
model checkpoint 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 predictions and inference.

HBox(children=(FloatProgress(value=0.0, description='Downloading', max=231508.0, style=Progress

HBox(children=(FloatProgress(value=0.0, description='Downloading', max=28.0, style=ProgressSty

HBox(children=(FloatProgress(value=0.0, description='Downloading', max=466062.0, style=Progress
```

[ ]:
```
model.summary()
```

```
Model: "tf_bert_for_sequence_classification"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bert (TFBertMainLayer)       multiple                  109482240
_________________________________________________________________
dropout_37 (Dropout)         multiple                  0
_________________________________________________________________
classifier (Dense)           multiple                  1538
=================================================================
Total params: 109,483,778
Trainable params: 109,483,778
Non-trainable params: 0
_________________________________________________________________
```

[ ]:
```
import tensorflow as tf
import pandas as pd
import os
import shutil
```

1.2 Loading Dataset

[ ]:
```
URL = "https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz"

dataset = tf.keras.utils.get_file(fname="aclImdb_v1.tar.gz",
                                  origin=URL,
                                  untar=True,
                                  cache_dir='.',
                                  cache_subdir='')
```

```
Downloading data from
https://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz
84131840/84125825 [==============================] - 1s 0us/step
```

[ ]:
```
main_dir = os.path.join(os.path.dirname(dataset), 'aclImdb')
train_dir = os.path.join(main_dir, 'train')
remove_dir = os.path.join(train_dir, 'unsup')
shutil.rmtree(remove_dir)
print(os.listdir(train_dir))
```

```
['urls_unsup.txt', 'urls_pos.txt', 'urls_neg.txt', 'pos', 'neg',
'labeledBow.feat', 'unsupBow.feat']
```

1.3 Splitting Dataset

[ ]:
```
train = tf.keras.preprocessing.text_dataset_from_directory(
    'aclImdb/train', batch_size=30000, validation_split=0.2,
    subset='training', seed=123)
test = tf.keras.preprocessing.text_dataset_from_directory(
    'aclImdb/train', batch_size=30000, validation_split=0.2,
    subset='validation', seed=123)
```

```
Found 25000 files belonging to 2 classes.
Using 20000 files for training.
Found 25000 files belonging to 2 classes.
Using 5000 files for validation.
```

1.4 Visualization

[ ]:
```
for i in train.take(1):
  train_feat = i[0].numpy()
  train_lab = i[1].numpy()

train = pd.DataFrame([train_feat, train_lab]).T
train.columns = ['DATA_COLUMN', 'LABEL_COLUMN']
train['DATA_COLUMN'] = train['DATA_COLUMN'].str.decode("utf-8")
train.head()
```

[ ]:

	DATA_COLUMN	LABEL_COLUMN
0	Canadian director Vincenzo Natali took the art...	1
1	I gave this film 10 not because it is a superb...	1
2	I admit to being somewhat jaded about the movi...	1
3	For a long time, 'The Menagerie' was my favori...	1
4	A truly frightening film. Feels as if it were ...	0

[ ]:
```
for j in test.take(1):
  test_feat = j[0].numpy()
  test_lab = j[1].numpy()

test = pd.DataFrame([test_feat, test_lab]).T
test.columns = ['DATA_COLUMN', 'LABEL_COLUMN']
test['DATA_COLUMN'] = test['DATA_COLUMN'].str.decode("utf-8")
test.head()
```

[ ]:

	DATA_COLUMN	LABEL_COLUMN
0	I can't believe that so much talent can be was...	0
1	This movie blows - let's get that straight rig...	0
2	The saddest thing about this "tribute" is that...	0
3	I'm only rating this film as a 3 out of pity b...	0
4	Something surprised me about this movie - it w...	1

[ ]:
```
def convert_data_to_examples(train, test, DATA_COLUMN, LABEL_COLUMN):
  train_InputExamples = train.apply(lambda x: InputExample(guid=None, #␣
 ↪Globally unique ID for bookkeeping, unused in this case
                                                          text_a =␣
 ↪x[DATA_COLUMN],
                                                          text_b = None,
                                                          label =␣
 ↪x[LABEL_COLUMN]), axis = 1)

  validation_InputExamples = test.apply(lambda x: InputExample(guid=None, #␣
 ↪Globally unique ID for bookkeeping, unused in this case
                                                          text_a =␣
 ↪x[DATA_COLUMN],
                                                          text_b = None,
                                                          label =␣
 ↪x[LABEL_COLUMN]), axis = 1)

  return train_InputExamples, validation_InputExamples

  train_InputExamples, validation_InputExamples =␣
 ↪convert_data_to_examples(train,
                                                                      test,
                                                                     ␣
 ↪'DATA_COLUMN',
```

```
␣
↪'LABEL_COLUMN')

def convert_examples_to_tf_dataset(examples, tokenizer, max_length=128):
    features = [] # -> will hold InputFeatures to be converted later

    for e in examples:
        # Documentation is really strong for this method, so please take a look␣
↪at it
        input_dict = tokenizer.encode_plus(
            e.text_a,
            add_special_tokens=True,
            max_length=max_length, # truncates if len(s) > max_length
            return_token_type_ids=True,
            return_attention_mask=True,
            pad_to_max_length=True, # pads to the right by default # CHECK THIS␣
↪for pad_to_max_length
            truncation=True
        )

        input_ids, token_type_ids, attention_mask = (input_dict["input_ids"],
            input_dict["token_type_ids"], input_dict['attention_mask'])

        features.append(
            InputFeatures(
                input_ids=input_ids, attention_mask=attention_mask,␣
↪token_type_ids=token_type_ids, label=e.label
            )
        )

    def gen():
        for f in features:
            yield (
                {
                    "input_ids": f.input_ids,
                    "attention_mask": f.attention_mask,
                    "token_type_ids": f.token_type_ids,
                },
                f.label,
            )

    return tf.data.Dataset.from_generator(
        gen,
        ({"input_ids": tf.int32, "attention_mask": tf.int32, "token_type_ids":␣
↪tf.int32}, tf.int64),
        (
            {
```

```
                "input_ids": tf.TensorShape([None]),
                "attention_mask": tf.TensorShape([None]),
                "token_type_ids": tf.TensorShape([None]),
            },
            tf.TensorShape([]),
        ),
    )


DATA_COLUMN = 'DATA_COLUMN'
LABEL_COLUMN = 'LABEL_COLUMN'
```

[ ]:
```
train_InputExamples, validation_InputExamples = convert_data_to_examples(train,␣
 ↪test, DATA_COLUMN, LABEL_COLUMN)

train_data = convert_examples_to_tf_dataset(list(train_InputExamples),␣
 ↪tokenizer)
train_data = train_data.shuffle(100).batch(32).repeat(2)

validation_data =␣
 ↪convert_examples_to_tf_dataset(list(validation_InputExamples), tokenizer)
validation_data = validation_data.batch(32)
```

```
/usr/local/lib/python3.7/dist-
packages/transformers/tokenization_utils_base.py:2074: FutureWarning: The
`pad_to_max_length` argument is deprecated and will be removed in a future
version, use `padding=True` or `padding='longest'` to pad to the longest
sequence in the batch, or use `padding='max_length'` to pad to a max length. In
this case, you can give a specific length with `max_length` (e.g.
`max_length=45`) or leave max_length to None to pad to the maximal input size of
the model (e.g. 512 for Bert).
  FutureWarning,
```

[ ]:
```
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=3e-5,␣
 ↪epsilon=1e-08, clipnorm=1.0),
              loss=tf.keras.losses.
 ↪SparseCategoricalCrossentropy(from_logits=True),
              metrics=[tf.keras.metrics.SparseCategoricalAccuracy('accuracy')])

model.fit(train_data, epochs=2, validation_data=validation_data)
```

```
Epoch 1/2
WARNING:tensorflow:The parameters `output_attentions`, `output_hidden_states`
and `use_cache` cannot be updated when calling a model.They have to be set to
True/False in the config object (i.e.: `config=XConfig.from_pretrained('name',
output_attentions=True)`).
WARNING:tensorflow:AutoGraph could not transform <bound method Socket.send of
```

```
<zmq.sugar.socket.Socket object at 0x7f2f39bd1d70>> and will run it as-is.
Please report this to the TensorFlow team. When filing the bug, set the
verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full
output.
Cause: module, class, method, function, traceback, frame, or code object was
expected, got cython_function_or_method
To silence this warning, decorate the function with
@tf.autograph.experimental.do_not_convert
WARNING: AutoGraph could not transform <bound method Socket.send of
<zmq.sugar.socket.Socket object at 0x7f2f39bd1d70>> and will run it as-is.
Please report this to the TensorFlow team. When filing the bug, set the
verbosity to 10 (on Linux, `export AUTOGRAPH_VERBOSITY=10`) and attach the full
output.
Cause: module, class, method, function, traceback, frame, or code object was
expected, got cython_function_or_method
To silence this warning, decorate the function with
@tf.autograph.experimental.do_not_convert
WARNING:tensorflow:AutoGraph could not transform <function wrap at
0x7f2f5547ec20> and will run it as-is.
Cause: while/else statement not yet supported
To silence this warning, decorate the function with
@tf.autograph.experimental.do_not_convert
WARNING: AutoGraph could not transform <function wrap at 0x7f2f5547ec20> and
will run it as-is.
Cause: while/else statement not yet supported
To silence this warning, decorate the function with
@tf.autograph.experimental.do_not_convert
WARNING:tensorflow:The parameter `return_dict` cannot be set in graph mode and
will always be set to `True`.
WARNING:tensorflow:The parameters `output_attentions`, `output_hidden_states`
and `use_cache` cannot be updated when calling a model.They have to be set to
True/False in the config object (i.e.: `config=XConfig.from_pretrained('name',
output_attentions=True)`).
WARNING:tensorflow:The parameter `return_dict` cannot be set in graph mode and
will always be set to `True`.
   1250/Unknown - 1096s 834ms/step - loss: 0.3596 - accuracy:
0.8333WARNING:tensorflow:The parameters `output_attentions`,
`output_hidden_states` and `use_cache` cannot be updated when calling a
model.They have to be set to True/False in the config object (i.e.:
`config=XConfig.from_pretrained('name', output_attentions=True)`).
WARNING:tensorflow:The parameter `return_dict` cannot be set in graph mode and
will always be set to `True`.
1250/1250 [==============================] - 1142s 871ms/step - loss: 0.3595 -
accuracy: 0.8333 - val_loss: 0.3481 - val_accuracy: 0.8734
Epoch 2/2
1250/1250 [==============================] - 1086s 869ms/step - loss: 0.0971 -
accuracy: 0.9661 - val_loss: 0.4237 - val_accuracy: 0.8824
```

[ ]:
```
<tensorflow.python.keras.callbacks.History at 0x7f2df7e9be90>
```

1.5 Evaluation

Highest Accuracy achieved on test set was 88.24%

[1]:
```
%cd drive/My\ Drive/
!pwd
```

```
/content/drive/My Drive
/content/drive/My Drive
```

[ ]:
```
!sudo apt-get install texlive-xetex texlive-fonts-recommended␣
↪texlive-generic-recommended
```

[ ]:
```
!jupyter nbconvert --to pdf Fuzzy_Systems_Twitter_V2.ipynb
```