In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score, f1_score
```

```
C:\Users\l-js\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning:
A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (d
etected version 1.24.3
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [2]:
```
train_df = pd.read_csv("train.csv")
test_df = pd.read_csv("test.csv")
sub_df = pd.read_csv("sampleSubmission.csv")
```

In [3]:
```
train_df.head()
```

Out[3]:

	PhraseId	SentenceId	Phrase	Sentiment
0	1	1	A series of escapades demonstrating the adage ...	1
1	2	1	A series of escapades demonstrating the adage ...	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2

In [4]:
```
test_df.head()
```

Out[4]:

	PhraseId	SentenceId	Phrase
0	156061	8545	An intermittently pleasing but mostly routine ...
1	156062	8545	An intermittently pleasing but mostly routine ...
2	156063	8545	An
3	156064	8545	intermittently pleasing but mostly routine effort
4	156065	8545	intermittently pleasing but mostly routine

In [5]:
```
sub_df.head()
```

Out[5]:

	PhraseId	Sentiment
0	156061	2
1	156062	2
2	156063	2
3	156064	2
4	156065	2

In [6]:
```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156060 entries, 0 to 156059
Data columns (total 4 columns):
 #   Column      Non-Null Count   Dtype 
---  ------      --------------   ----- 
 0    PhraseId   156060 non-null  int64 
 1   SentenceId  156060 non-null  int64 
 2   Phrase      156060 non-null  object
 3   Sentiment   156060 non-null  int64 
dtypes: int64(3), object(1)
memory usage: 4.8+ MB
```

In [7]:
```
print("train : ", train_df.shape)
print("test : ", test_df.shape)
print("submission : ", sub_df.shape)
```

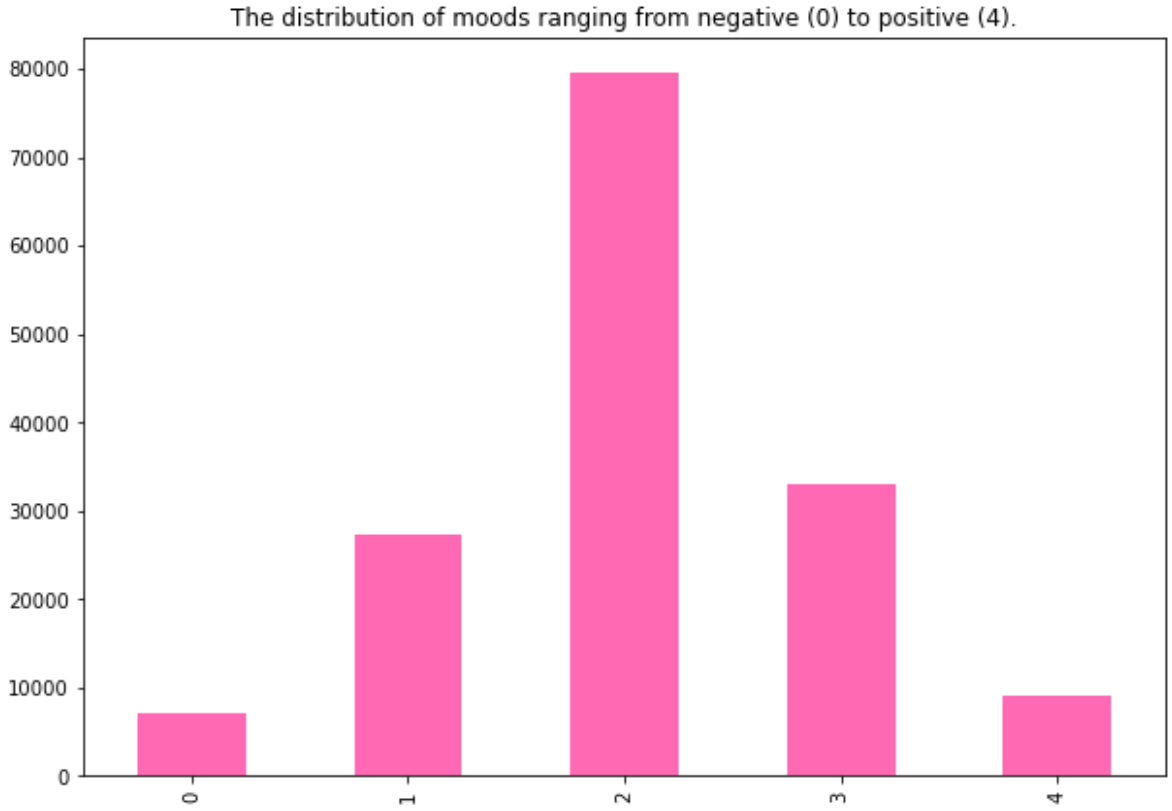
```
train :  (156060, 4)
test :  (66292, 3)
submission :  (66292, 2)
```

In [8]:
```
print(train_df.Sentiment.value_counts(normalize = True).sort_index())
```

```
0    0.045316
1    0.174760
2    0.509945
3    0.210989
4    0.058990
Name: Sentiment, dtype: float64
```

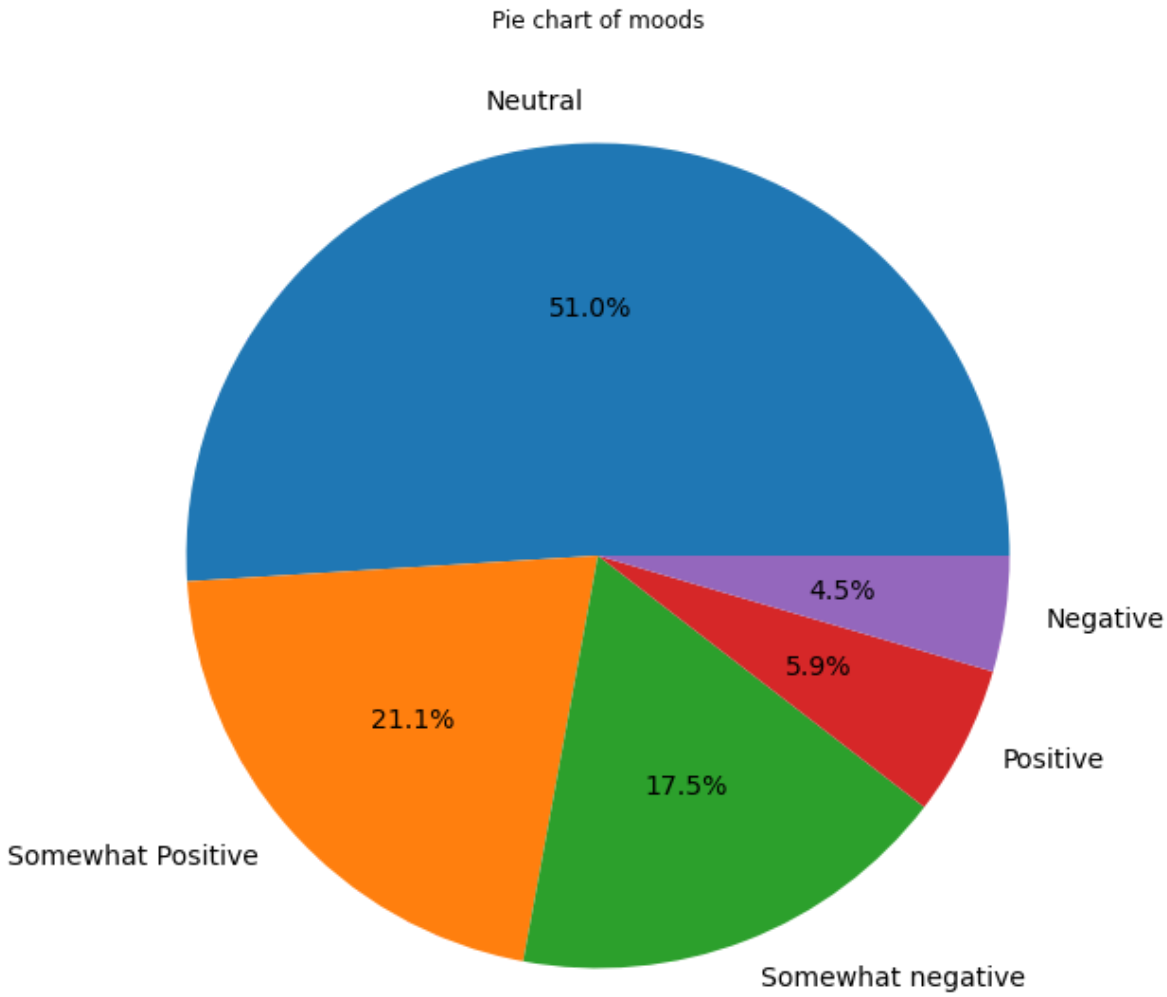
In [9]:
```
train_df.Sentiment.value_counts().sort_index().plot(kind = 'bar', color = 'hot
```

Out[9]:
```
<AxesSubplot:title={'center':'The distribution of moods ranging from negative
(0) to positive (4).'}>
```



In [10]:
```
df2 = train_df.copy(deep = True)
pie1 = pd.DataFrame(df2['Sentiment'].replace(0, 'Negative').replace(1, 'Somewh
pie1.reset_index(inplace = True)
pie1.plot(kind = 'pie', title = 'Pie chart of moods', y = 'Sentiment', autopct
```

Out[10]: <AxesSubplot:title={'center':'Pie chart of moods'}>



In [11]:
```
train_df.Phrase.sample(10).values
```

Out[11]:
```
array(["you wo n't be disappointed", 'a hilarious ode', 'circles',
       'worthy of her considerable talents', 'in the fifth Trek flick',
       'pleasuring', 'is meaningless , vapid and devoid of substance ,',
       'obviously aimed at kids', 'Tennessee Williams', 'a great'],
      dtype=object)
```

In [12]:
```
train_df.Phrase
```

Out[12]:
```
0         A series of escapades demonstrating the adage ...
1         A series of escapades demonstrating the adage ...
2                                                 A series
3                                                        A
4                                                   series
                              ...
156055                                           Hearst 's
156056                    forced avuncular chortles
156057                           avuncular chortles
156058                                    avuncular
156059                                     chortles
Name: Phrase, Length: 156060, dtype: object
```

In [13]:
```
train_df[train_df.Sentiment == 0].Phrase.values[:10]
```

Out[13]:
```
array(['would have a hard time sitting through this one',
       'have a hard time sitting through this one',
       'Aggressive self-glorification and a manipulative whitewash',
       'self-glorification and a manipulative whitewash',
       'Trouble Every Day is a plodding mess .', 'is a plodding mess',
       'plodding mess', 'could hate it for the same reason', 'hate it',
       'hate'], dtype=object)
```

In [14]:
```
train_df[train_df.Sentiment == 1].Phrase.values[:10]
```

Out[14]:
```
array(['A series of escapades demonstrating the adage that what is good for t
he goose is also good for the gander , some of which occasionally amuses but
none of which amounts to much of a story .',
       'the gander , some of which occasionally amuses but none of which amou
nts to much of a story',
       'but none of which amounts to much of a story',
       'none of which amounts to much of a story',
       "Even fans of Ismail Merchant 's work , I suspect , would have a hard
time sitting through this one .",
       ', I suspect , would have a hard time sitting through this one .',
       'would have a hard time sitting through this one .',
       'a hard time sitting through this one', 'a hard time', 'hard time'],
      dtype=object)
```

In [15]:
```
train_df[train_df.Sentiment == 2].Phrase.values[:10]
```

Out[15]:
```
array(['A series of escapades demonstrating the adage that what is good for t
he goose',
       'A series', 'A', 'series',
       'of escapades demonstrating the adage that what is good for the goos
e',
       'of',
       'escapades demonstrating the adage that what is good for the goose',
       'escapades',
       'demonstrating the adage that what is good for the goose',
       'demonstrating the adage'], dtype=object)
```

In [16]:
```
train_df[train_df.Sentiment == 3].Phrase.values[:10]
```

Out[16]:
```
array(['good for the goose', 'good', 'amuses',
       'This quiet , introspective and entertaining independent',
       'quiet , introspective and entertaining',
       ', introspective and entertaining',
       'introspective and entertaining', 'introspective and',
       'is worth seeking .', 'fans'], dtype=object)
```

In [17]:
```
train_df[train_df.Sentiment == 4].Phrase.values[:10]
```

Out[17]:
```
array(['This quiet , introspective and entertaining independent is worth seek
ing .',
       'quiet , introspective and entertaining independent',
       'entertaining', 'is worth seeking',
       'A positively thrilling combination of ethnography and all the intrigu
e , betrayal , deceit and murder of a Shakespearean tragedy or a juicy soap o
pera',
       'A positively thrilling combination of ethnography and all the intrigu
e , betrayal , deceit and murder',
       'thrilling',
       'A comedy-drama of nearly epic proportions rooted in a sincere perform
ance by the title character undergoing midlife crisis .',
       'nearly epic',
       'rooted in a sincere performance by the title character undergoing mid
life crisis .'],
      dtype=object)
```

In [18]:
```
train_df.shape, test_df.shape
```

Out[18]:
```
((156060, 4), (66292, 3))
```

In [19]:
```
import nltk
from nltk.tokenize import word_tokenize
from nltk.stem.snowball import SnowballStemmer
from nltk.corpus import stopwords
```

In [20]:
```
stemmer  = SnowballStemmer(language='english')
```

In [21]:
```
seq_len = 512
num_samples = len(train_df)
num_samples
```

Out[21]:
```
156060
```

In [22]:
```
pip install transformers
```

```
Requirement already satisfied: transformers in c:\users\l-js\anaconda3\lib\si
te-packages (4.31.0)
Requirement already satisfied: tqdm>=4.27 in c:\users\l-js\anaconda3\lib\site
-packages (from transformers) (4.62.3)
Requirement already satisfied: pyyaml>=5.1 in c:\users\l-js\anaconda3\lib\sit
e-packages (from transformers) (6.0)
Requirement already satisfied: numpy>=1.17 in c:\users\l-js\anaconda3\lib\sit
e-packages (from transformers) (1.24.3)
Requirement already satisfied: tokenizers!=0.11.3,<0.14,>=0.11.1 in c:\users
\l-js\anaconda3\lib\site-packages (from transformers) (0.13.3)
Requirement already satisfied: requests in c:\users\l-js\anaconda3\lib\site-p
ackages (from transformers) (2.26.0)
Requirement already satisfied: regex!=2019.12.17 in c:\users\l-js\anaconda3\l
ib\site-packages (from transformers) (2021.8.3)
Requirement already satisfied: filelock in c:\users\l-js\anaconda3\lib\site-p
ackages (from transformers) (3.3.1)
Requirement already satisfied: safetensors>=0.3.1 in c:\users\l-js\anaconda3
\lib\site-packages (from transformers) (0.3.2)
Requirement already satisfied: huggingface-hub<1.0,>=0.14.1 in c:\users\l-js
\anaconda3\lib\site-packages (from transformers) (0.16.4)
Requirement already satisfied: packaging>=20.0 in c:\users\l-js\anaconda3\lib
\site-packages (from transformers) (21.0)
Requirement already satisfied: fsspec in c:\users\l-js\anaconda3\lib\site-pac
kages (from huggingface-hub<1.0,>=0.14.1->transformers) (2021.10.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\l-js\an
aconda3\lib\site-packages (from huggingface-hub<1.0,>=0.14.1->transformers)
(3.10.0.2)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\l-js\anaconda3\li
b\site-packages (from packaging>=20.0->transformers) (3.0.4)
Requirement already satisfied: colorama in c:\users\l-js\anaconda3\lib\site-p
ackages (from tqdm>=4.27->transformers) (0.4.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\l-js\anaconda3\lib\si
te-packages (from requests->transformers) (3.2)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\l-js\anacond
a3\lib\site-packages (from requests->transformers) (1.26.7)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\l-js\anaconda3
\lib\site-packages (from requests->transformers) (2021.10.8)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\l-js\ana
conda3\lib\site-packages (from requests->transformers) (2.0.4)
Note: you may need to restart the kernel to use updated packages.
```

In [23]:
```
from transformers import BertTokenizer
```

In [24]:
```
tokenizer = BertTokenizer.from_pretrained("bert-base-cased")
```

In [25]:
```
tokens = tokenizer(train_df["Phrase"].tolist(), max_length = seq_len, truncati
```

In [26]:
```
tokens.keys()
```

Out[26]:
```
dict_keys(['input_ids', 'token_type_ids', 'attention_mask'])
```

In [27]:
```
tokens["input_ids"], tokens["attention_mask"]
```

Out[27]:
```
(array([[  101,   138,  1326, ...,     0,     0,     0],
        [  101,   138,  1326, ...,     0,     0,     0],
        [  101,   138,  1326, ...,     0,     0,     0],
        ...,
        [  101,   170, 25247, ...,     0,     0,     0],
        [  101,   170, 25247, ...,     0,     0,     0],
        [  101, 22572, 12148, ...,     0,     0,     0]]),
 array([[1, 1, 1, ..., 0, 0, 0],
        [1, 1, 1, ..., 0, 0, 0],
        [1, 1, 1, ..., 0, 0, 0],
        ...,
        [1, 1, 1, ..., 0, 0, 0],
        [1, 1, 1, ..., 0, 0, 0],
        [1, 1, 1, ..., 0, 0, 0]]))
```

In [28]:
```
classes_arr = train_df["Sentiment"].values
classes_arr
```

Out[28]:
```
array([1, 2, 2, ..., 3, 2, 2], dtype=int64)
```

In [29]:
```
import numpy as np
labels = np.zeros((num_samples, classes_arr.max()+1))
labels.shape
```

Out[29]:
```
(156060, 5)
```

In [30]:
```
labels[np.arange(num_samples), classes_arr] = 1
labels
```

Out[30]:
```
array([[0., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 1., 0., 0.],
       ...,
       [0., 0., 0., 1., 0.],
       [0., 0., 1., 0., 0.],
       [0., 0., 1., 0., 0.]])
```

In [32]:
```
pip install tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\l-js\anaconda3\lib\s
ite-packages (2.13.0)
Requirement already satisfied: tensorflow-intel==2.13.0 in c:\users\l-js\a
naconda3\lib\site-packages (from tensorflow) (2.13.0)
Requirement already satisfied: keras<2.14,>=2.13.1 in c:\users\l-js\anacon
da3\lib\site-packages (from tensorflow-intel==2.13.0->tensorflow) (2.13.1)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\l-js\anaconda
3\lib\site-packages (from tensorflow-intel==2.13.0->tensorflow) (1.6.3)
Requirement already satisfied: tensorflow-estimator<2.14,>=2.13.0 in c:\us
ers\l-js\anaconda3\lib\site-packages (from tensorflow-intel==2.13.0->tenso
rflow) (2.13.0)
Requirement already satisfied: six>=1.12.0 in c:\users\l-js\anaconda3\lib
\site-packages (from tensorflow-intel==2.13.0->tensorflow) (1.16.0)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\l-js\anaconda3\li
b\site-packages (from tensorflow-intel==2.13.0->tensorflow) (1.12.1)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in c:\users\l-js\anacon
da3\lib\site-packages (from tensorflow-intel==2.13.0->tensorflow) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\l-js\anacon
da3\lib\site-packages (from tensorflow-intel==2.13.0->tensorflow) (0.2.0)
```

In [33]:
```
import tensorflow as tf
```

In [34]:
```
dataset = tf.data.Dataset.from_tensor_slices((tokens["input_ids"],tokens["atte
```

In [35]:
```
dataset.take(1)
```

Out[35]:
```
<_TakeDataset element_spec=(TensorSpec(shape=(512,), dtype=tf.int32, name=Non
e), TensorSpec(shape=(512,), dtype=tf.int32, name=None), TensorSpec(shape=
(5,), dtype=tf.float64, name=None))>
```

In [37]:
```
def map_func(input_ids, masks, labels):
    mask = tf.math.not_equal(input_ids, 0)

    return{'input_ids': input_ids, 'attention_mask': mask}, labels
dataset = dataset.map(map_func)
dataset.take(1)
```

Out[37]:
```
<_TakeDataset element_spec=({'input_ids': TensorSpec(shape=(512,), dtype=tf.i
nt32, name=None), 'attention_mask': TensorSpec(shape=(512,), dtype=tf.bool, n
ame=None)}, TensorSpec(shape=(5,), dtype=tf.float64, name=None))>
```

In [38]:
```
batch_size = 16
dataset = dataset.shuffle(10000).batch(batch_size, drop_remainder = True)
```

In [39]:
```
dataset.take(1)
```

Out[39]:
```
<_TakeDataset element_spec=({'input_ids': TensorSpec(shape=(16, 512), dtype=t
f.int32, name=None), 'attention_mask': TensorSpec(shape=(16, 512), dtype=tf.b
ool, name=None)}, TensorSpec(shape=(16, 5), dtype=tf.float64, name=None))>
```

In [40]:
```
split = 0.9
size = int((tokens['input_ids'].shape[0]/ batch_size)*split)
size
```

Out[40]:
```
8778
```

In [42]:
```
train_ds = dataset.take(size)
val_ds = dataset.skip(size)
```

In [43]:
```
train_ds.take(1)
```

Out[43]:
```
<_TakeDataset element_spec=({'input_ids': TensorSpec(shape=(16, 512), dtype=t
f.int32, name=None), 'attention_mask': TensorSpec(shape=(16, 512), dtype=tf.b
ool, name=None)}, TensorSpec(shape=(16, 5), dtype=tf.float64, name=None))>
```

In [44]:
```
from transformers import TFAutoModel
```

In [46]:
```
bert = TFAutoModel.from_pretrained("bert-base-cased")
```

```
Downloading                                          436M/436M [01:40<00:00,
model.safetensors: 100%                              4.34MB/s]

C:\Users\l-js\anaconda3\lib\site-packages\huggingface_hub\file_download.py:133: UserWarning: `huggingface_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not support them in C:\Users\l-js\.cache\huggingface\hub. Caching files will still work but in a degraded version that might require more space on your disk. This warning can be disabled by setting the `HF_HUB_DISABLE_SYMLINKS_WARNING` environment variable. For more details, see https://huggingface.co/docs/huggingface_hub/how-to-cache#limitations. (https://huggingface.co/docs/huggingface_hub/how-to-cache#limitations.)
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to see activate developer mode, see this article: https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development (https://docs.microsoft.com/en-us/windows/apps/get-started/enable-your-device-for-development)
  warnings.warn(message)
Some weights of the PyTorch model were not used when initializing the TF 2.0 model TFBertModel: ['cls.predictions.transform.dense.bias', 'cls.predictions.transform.dense.weight', 'cls.predictions.bias', 'cls.seq_relationship.bias', 'cls.predictions.transform.LayerNorm.bias', 'cls.seq_relationship.weight', 'cls.predictions.transform.LayerNorm.weight']
- This IS expected if you are initializing TFBertModel from a PyTorch model trained on another task or with another architecture (e.g. initializing a TFBertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing TFBertModel from a PyTorch model that you expect to be exactly identical (e.g. initializing a TFBertForSequenceClassification model from a BertForSequenceClassification model).
All the weights of TFBertModel were initialized from the PyTorch model.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFBertModel for predictions without further training.
```

In [47]:
```
bert.summary()
```

```
Model: "tf_bert_model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bert (TFBertMainLayer)      multiple                  108310272

=================================================================
Total params: 108310272 (413.17 MB)
Trainable params: 108310272 (413.17 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [48]:
```
input_ids = tf.keras.layers.Input(shape=(512,), name='input_ids', dtype='int32
mask = tf.keras.layers.Input(shape=(512,), name='attention_mask', dtype='int32
embeddings = bert.bert(input_ids, attention_mask=mask)[1]
x = tf.keras.layers.Dense(1024, activation='relu')(embeddings)
y = tf.keras.layers.Dense(5, activation='softmax', name='outputs')(x)
```

In [49]:
```
model = tf.keras.Model(inputs=[input_ids, mask], outputs=y)
model.layers[2].trainable = False
model.summary()
```

```
Model: "model"
_________________________________________________________________________________
_____________________
 Layer (type)                Output Shape              Param #   Connected
to
==================================================================================
====================
 input_ids (InputLayer)      [(None, 512)]             0         []

 attention_mask (InputLayer  [(None, 512)]             0         []
 )

 bert (TFBertMainLayer)      TFBaseModelOutputWithPooli   1083102   ['input_i
ds[0][0]',
                             ngAndCrossAttentions(last_   72         'attenti
on_mask[0][0]']
                             hidden_state=(None, 512, 7
                             68),
                              pooler_output=(None, 768)
                             , past_key_values=None, hi
                             dden_states=None, attentio
                             ns=None, cross_attentions=
                             None)

 dense (Dense)               (None, 1024)              787456    ['bert[0]
[1]']

 outputs (Dense)             (None, 5)                 5125      ['dense
[0][0]']

==================================================================================
====================
Total params: 109102853 (416.19 MB)
Trainable params: 792581 (3.02 MB)
Non-trainable params: 108310272 (413.17 MB)
_________________________________________________________________________________
_____________________
```

In [53]:
```
optimizer = tf.keras.optimizers.Adam(learning_rate=1e-5)
loss = tf.keras.losses.CategoricalCrossentropy()
acc = tf.keras.metrics.CategoricalAccuracy('accuracy')

model.compile(optimizer=optimizer, loss=loss, metrics=[acc])
```

In [*]:
```
history = model.fit(train_ds, validation_data=val_ds, epochs=1)
```

```
 160/8778 [..............................] - ETA: 76:28:04 - loss: 1.1968 - a
ccuracy: 0.5660
```