In [1]:

```
import torch

# If there's a GPU available...
if torch.cuda.is_available():

    # Tell PyTorch to use the GPU.
    device = torch.device("cuda")

    print('There are %d GPU(s) available.' % torch.cuda.device_count())

    print('We will use the GPU:', torch.cuda.get_device_name(0))
    !nvidia-smi

# If not...
else:
    print('No GPU available, using the CPU instead.')
    device = torch.device("cpu")
```

```
There are 1 GPU(s) available.
We will use the GPU: Tesla T4
Fri Jun 10 15:22:35 2022
+---------------------------------------------------------------------
----------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 
11.2     |
|-------------------------------+----------------------+------------
----------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Un
corr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  C
ompute M. |
|                               |                      |            
MIG M. |
|===============================+======================+============
==========|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |            
0 |
| N/A   59C    P8    15W /  70W |      3MiB / 15109MiB |      0%    
Default |
|                               |                      |            
N/A |
+-------------------------------+----------------------+------------
----------+

+---------------------------------------------------------------------
----------+
| Processes:                                                          
|
|  GPU   GI   CI        PID   Type   Process name                  G
PU Memory |
|        ID   ID                                                   U
sage      |
|=====================================================================
==========|
|  No running processes found                                         
|
+---------------------------------------------------------------------
----------+
```

Importing the libraries needed

In [2]:

```
!pip install pyarabic
!pip install emoji
!pip install pystemmer
!pip install optuna==2.3.0
!pip install transformers==4.2.1
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.d
ev/colab-wheels/public/simple/
Collecting pyarabic
  Downloading PyArabic-0.6.14-py3-none-any.whl (126 kB)
     |████████████████████████████████| 126 kB 13.8 MB/s
Requirement already satisfied: six>=1.14.0 in /usr/local/lib/python
3.7/dist-packages (from pyarabic) (1.15.0)
Installing collected packages: pyarabic
Successfully installed pyarabic-0.6.14
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.d
ev/colab-wheels/public/simple/
Collecting emoji
  Downloading emoji-1.7.0.tar.gz (175 kB)
     |████████████████████████████████| 175 kB 15.6 MB/s
Building wheels for collected packages: emoji
  Building wheel for emoji (setup.py) ... done
  Created wheel for emoji: filename=emoji-1.7.0-py3-none-any.whl siz
e=171046 sha256=ad7285bcc10990b894ab1be7949d3210dd3b7bb6dd99ae5d7786
ae30a3bbbd67
  Stored in directory: /root/.cache/pip/wheels/8a/4e/b6/57b01db010d1
7ef6ea9b40300af725ef3e210cb1acfb7ac8b6
Successfully built emoji
Installing collected packages: emoji
Successfully installed emoji-1.7.0
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.d
ev/colab-wheels/public/simple/
Collecting pystemmer
  Downloading PyStemmer-2.0.1.tar.gz (559 kB)
     |████████████████████████████████| 559 kB 14.6 MB/s
Building wheels for collected packages: pystemmer
  Building wheel for pystemmer (setup.py) ... done
  Created wheel for pystemmer: filename=PyStemmer-2.0.1-cp37-cp37m-l
inux_x86_64.whl size=425684 sha256=74bca66e15275ac733fa8ba882c1c3437
ab1e9a3b4b29a28aeb45d36d1d1dba3
  Stored in directory: /root/.cache/pip/wheels/30/6d/40/0d17a498c500
9922dbb3ddaca3d3652387ba94cc96142001f0
Successfully built pystemmer
Installing collected packages: pystemmer
Successfully installed pystemmer-2.0.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.d
ev/colab-wheels/public/simple/
Collecting optuna==2.3.0
  Downloading optuna-2.3.0.tar.gz (258 kB)
     |████████████████████████████████| 258 kB 14.2 MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
    Preparing wheel metadata ... done
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dis
t-packages (from optuna==2.3.0) (1.21.6)
Collecting colorlog
  Downloading colorlog-6.6.0-py2.py3-none-any.whl (11 kB)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist
-packages (from optuna==2.3.0) (4.64.0)
Requirement already satisfied: sqlalchemy>=1.1.0 in /usr/local/lib/p
ython3.7/dist-packages (from optuna==2.3.0) (1.4.36)
Requirement already satisfied: scipy!=1.4.0 in /usr/local/lib/python
3.7/dist-packages (from optuna==2.3.0) (1.4.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/pyt
hon3.7/dist-packages (from optuna==2.3.0) (21.3)
Collecting cliff
  Downloading cliff-3.10.1-py3-none-any.whl (81 kB)
```

```
     |████████████████████████████████| 81 kB 11.2 MB/s
Collecting alembic
  Downloading alembic-1.8.0-py3-none-any.whl (209 kB)
     |████████████████████████████████| 209 kB 75.4 MB/s
Collecting cmaes>=0.6.0
  Downloading cmaes-0.8.2-py3-none-any.whl (15 kB)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/di
st-packages (from optuna==2.3.0) (1.1.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/loca
l/lib/python3.7/dist-packages (from packaging>=20.0->optuna==2.3.0)
(3.0.9)
Requirement already satisfied: importlib-metadata in /usr/local/lib/
python3.7/dist-packages (from sqlalchemy>=1.1.0->optuna==2.3.0) (4.1
1.4)
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/py
thon3.7/dist-packages (from sqlalchemy>=1.1.0->optuna==2.3.0) (1.1.
2)
Collecting Mako
  Downloading Mako-1.2.0-py3-none-any.whl (78 kB)
     |████████████████████████████████| 78 kB 9.6 MB/s
Requirement already satisfied: importlib-resources in /usr/local/li
b/python3.7/dist-packages (from alembic->optuna==2.3.0) (5.7.1)
Collecting stevedore>=2.0.1
  Downloading stevedore-3.5.0-py3-none-any.whl (49 kB)
     |████████████████████████████████| 49 kB 8.4 MB/s
Collecting pbr!=2.1.0,>=2.0.0
  Downloading pbr-5.9.0-py2.py3-none-any.whl (112 kB)
     |████████████████████████████████| 112 kB 75.7 MB/s
Collecting cmd2>=1.0.0
  Downloading cmd2-2.4.1-py3-none-any.whl (146 kB)
     |████████████████████████████████| 146 kB 72.0 MB/s
Requirement already satisfied: PrettyTable>=0.7.2 in /usr/local/lib/
python3.7/dist-packages (from cliff->optuna==2.3.0) (3.3.0)
Collecting autopage>=0.4.0
  Downloading autopage-0.5.1-py3-none-any.whl (29 kB)
Requirement already satisfied: PyYAML>=3.12 in /usr/local/lib/python
3.7/dist-packages (from cliff->optuna==2.3.0) (3.13)
Requirement already satisfied: wcwidth>=0.1.7 in /usr/local/lib/pyth
on3.7/dist-packages (from cmd2>=1.0.0->cliff->optuna==2.3.0) (0.2.5)
Requirement already satisfied: typing-extensions in /usr/local/lib/p
ython3.7/dist-packages (from cmd2>=1.0.0->cliff->optuna==2.3.0) (4.
2.0)
Requirement already satisfied: attrs>=16.3.0 in /usr/local/lib/pytho
n3.7/dist-packages (from cmd2>=1.0.0->cliff->optuna==2.3.0) (21.4.0)
Collecting pyperclip>=1.6
  Downloading pyperclip-1.8.2.tar.gz (20 kB)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.
7/dist-packages (from importlib-metadata->sqlalchemy>=1.1.0->optuna=
=2.3.0) (3.8.0)
Requirement already satisfied: MarkupSafe>=0.9.2 in /usr/local/lib/p
ython3.7/dist-packages (from Mako->alembic->optuna==2.3.0) (2.0.1)
Building wheels for collected packages: optuna, pyperclip
  Building wheel for optuna (PEP 517) ... done
  Created wheel for optuna: filename=optuna-2.3.0-py3-none-any.whl s
ize=359772 sha256=d4e78ff9980ee309d23f08594bd36e299c9740b3b81f1f655a
b729ef9e2bbac3
  Stored in directory: /root/.cache/pip/wheels/38/61/9e/955ab1890f6c
ab231b1d756db63f36c711968a324296e0b649
  Building wheel for pyperclip (setup.py) ... done
  Created wheel for pyperclip: filename=pyperclip-1.8.2-py3-none-an
y.whl size=11137 sha256=9f2585c30df9021ceefd1ff5d4df6ef888a6ceb81401
```

```
c7f0aef8c07e423e4ed2
  Stored in directory: /root/.cache/pip/wheels/9f/18/84/8f69f8b08169
c7bae2dde6bd7daf0c19fca8c8e500ee620a28
Successfully built optuna pyperclip
Installing collected packages: pyperclip, pbr, stevedore, Mako, cmd
2, autopage, colorlog, cmaes, cliff, alembic, optuna
Successfully installed Mako-1.2.0 alembic-1.8.0 autopage-0.5.1 cliff
-3.10.1 cmaes-0.8.2 cmd2-2.4.1 colorlog-6.6.0 optuna-2.3.0 pbr-5.9.0
pyperclip-1.8.2 stevedore-3.5.0
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.d
ev/colab-wheels/public/simple/
Collecting transformers==4.2.1
  Downloading transformers-4.2.1-py3-none-any.whl (1.8 MB)
     |████████████████████████████████| 1.8 MB 14.9 MB/s
Collecting sacremoses
  Downloading sacremoses-0.0.53.tar.gz (880 kB)
     |████████████████████████████████| 880 kB 56.0 MB/s
Requirement already satisfied: filelock in /usr/local/lib/python3.7/
dist-packages (from transformers==4.2.1) (3.7.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dis
t-packages (from transformers==4.2.1) (1.21.6)
Requirement already satisfied: requests in /usr/local/lib/python3.7/
dist-packages (from transformers==4.2.1) (2.23.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.
7/dist-packages (from transformers==4.2.1) (21.3)
Requirement already satisfied: importlib-metadata in /usr/local/lib/
python3.7/dist-packages (from transformers==4.2.1) (4.11.4)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.
7/dist-packages (from transformers==4.2.1) (4.64.0)
Collecting tokenizers==0.9.4
  Downloading tokenizers-0.9.4-cp37-cp37m-manylinux2010_x86_64.whl
(2.9 MB)
     |████████████████████████████████| 2.9 MB 53.7 MB/s
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/p
ython3.7/dist-packages (from transformers==4.2.1) (2019.12.20)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.
7/dist-packages (from importlib-metadata->transformers==4.2.1) (3.8.
0)
Requirement already satisfied: typing-extensions>=3.6.4 in /usr/loca
l/lib/python3.7/dist-packages (from importlib-metadata->transformers
==4.2.1) (4.2.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/loca
l/lib/python3.7/dist-packages (from packaging->transformers==4.2.1)
(3.0.9)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/
python3.7/dist-packages (from requests->transformers==4.2.1) (2022.
5.18.1)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python
3.7/dist-packages (from requests->transformers==4.2.1) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/p
ython3.7/dist-packages (from requests->transformers==4.2.1) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.2
1.1 in /usr/local/lib/python3.7/dist-packages (from requests->transf
ormers==4.2.1) (1.24.3)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-
packages (from sacremoses->transformers==4.2.1) (1.15.0)
Requirement already satisfied: click in /usr/local/lib/python3.7/dis
t-packages (from sacremoses->transformers==4.2.1) (7.1.2)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/di
st-packages (from sacremoses->transformers==4.2.1) (1.1.0)
Building wheels for collected packages: sacremoses
```

```
  Building wheel for sacremoses (setup.py) ... done
  Created wheel for sacremoses: filename=sacremoses-0.0.53-py3-none-
any.whl size=895260 sha256=222df4bbe2cc9ced8f1e0974495b72d1670a89617
905a1a4dffc362494e8a512
  Stored in directory: /root/.cache/pip/wheels/87/39/dd/a83eeef36d0b
f98e7a4d1933a4ad2d660295a40613079bafc9
Successfully built sacremoses
Installing collected packages: tokenizers, sacremoses, transformers
Successfully installed sacremoses-0.0.53 tokenizers-0.9.4 transforme
rs-4.2.1
```

In [3]:

```
import numpy as np
import pandas as pd
import pyarabic.araby as ar

import re , emoji, Stemmer, functools, operator, string
import torch , optuna, gc, random, os

import matplotlib.pyplot as plt
import seaborn as sns

from tqdm import tqdm_notebook as tqdm
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score, f1_score, con
fusion_matrix, precision_score , recall_score
from transformers import AutoConfig, AutoModelForSequenceClassification, AutoTok
enizer
from transformers.data.processors import SingleSentenceClassificationProcessor
from transformers import Trainer , TrainingArguments
from transformers.trainer_utils import EvaluationStrategy
from transformers.data.processors.utils import InputFeatures
from torch.utils.data import Dataset
from torch.utils.data import DataLoader
from sklearn.utils import resample
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report, accuracy_sc
ore

import gensim
from gensim.models import KeyedVectors

from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

import tensorflow as tf
from keras.models import Sequential
from tensorflow.keras.layers import SpatialDropout1D, Conv1D, Bidirectional, LST
M, Dense, Input, Dropout, GlobalMaxPooling1D
from keras.layers.embeddings import Embedding
from tensorflow.keras.callbacks import ModelCheckpoint, ReduceLROnPlateau, Early
Stopping
from tensorflow.keras.optimizers import Adam

import itertools
from numpy import loadtxt
from keras.models import load_model

import logging

logging.basicConfig(level=logging.WARNING)
logger = logging.getLogger(__name__)
```

the function of the preprocessing

In [4]:

```python
st =  Stemmer.Stemmer('arabic')
def data_cleaning (x):
    x = re.sub('@[^\s]+', ' ', x)
    x = re.sub('((www\.[^\s]+)|(https?://[^\s]+))',' ',x)

    emoji_pattern = re.compile("["
                               u"\U0001F600-\U0001F64F"  # emoticons
                               u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                               u"\U0001F680-\U0001F6FF"  # transport & map symbo
ls
                               u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                               u"\U00002500-\U00002BEF"  # chinese char
                               u"\U00002702-\U000027B0"
                               u"\U00002702-\U000027B0"
                               u"\U000024C2-\U0001F251"
                               u"\U0001f926-\U0001f937"
                               u"\U00010000-\U0010ffff"
                               u"\u2640-\u2642"
                               u"\u2600-\u2B55"
                               u"\u200d"
                               u"\u23cf"
                               u"\u23e9"
                               u"\u231a"
                               u"\ufe0f"  # dingbats
                               u"\u3030""]+", flags=re.UNICODE)
    emoji_pattern.sub(r'', x)

    ar_punctuations = '''`÷×؛<>_()*&^%][ـ،/:"؟.,'{}~¦+|!”…“–ـ'''
    en_punctuations = string.punctuation
    punctuations = ar_punctuations + en_punctuations
    x = x.translate(str.maketrans('', '', punctuations))

    arabic_diacritics = re.compile(""" ّ    | # Tashdid
                             َ    | # Fatha
                             ً    | # Tanwin Fath
                             ُ    | # Damma
                             ٌ    | # Tanwin Damm
                             ِ    | # Kasra
                             ٍ    | # Tanwin Kasr
                             ْ    | # Sukun
                             ـ      # Tatwil/Kashida
                         """, re.VERBOSE)
    x = re.sub(arabic_diacritics, '', str(x))

#     x = re.sub("[إأآا]", "ا", x)
#     x = re.sub("ى", "ي", x)
#     x = re.sub("ة", "ه", x)
#     x = re.sub("گ", "ك", x)
#     x = re.sub(r'(.)\1+', r'\1', x)

    return x
```

Connecting to google drive

In [5]:

```
from google.colab import drive
drive.mount("/content/gdrive")
```

```
Mounted at /content/gdrive
```

Uploading the dataset

In [6]:

```
Text_Col_Train = "review"
Sentiment_Col_Train = "sentiment"
Train_Data_File = "/content/gdrive/MyDrive/thesis/HARD.xlsx"

train_data = pd.DataFrame()

train_data = pd.read_excel(Train_Data_File)

train_data.head(3)
```

Out[6]:

	no	Hotel name	rating	user type	room type	nights	review
0	2	فندق 72	2	مسافر منفرد	غرفة ديلوكس مزدوجة أو توأم	أقمت ليلة واحدة	“ممتاز”. النظافة والطاقم متعاون.
1	3	فندق 72	5	زوج	غرفة ديلوكس مزدوجة أو توأم	أقمت ليلة واحدة	استثنائي. سهولة إنهاء المعاملة في الاستقبال. ل...
2	16	فندق 72	5	زوج	-	أقمت ليلتين	استثنائي. انصح بأختيار الاسويت و بالاخص غرفه ر...

In [7]:

```
print(train_data.rating.value_counts())
```

```
2    38467
4    26450
5    26399
1    14382
Name: rating, dtype: int64
```

printing the fiels with missed values

In [8]:

```
train_data.isnull().sum()
```

Out[8]:

```
no            0
Hotel name    0
rating        0
user type     0
room type     0
nights        0
review        0
dtype: int64
```

printing the number of the duplicated rows

In [9]:

```
print("On a  {} doublons dans Data.".format(train_data.duplicated().sum()))
```

```
On a  0 doublons dans Data.
```

checking the types of the fiels in the data

In [10]:

```
train_data.dtypes
```

Out[10]:

```
no             int64
Hotel name    object
rating         int64
user type     object
room type     object
nights        object
review        object
dtype: object
```

function for printing the pie

In [11]:

```
def pie(data,col):
    labels = data[col].value_counts().keys().tolist()
    n = len(labels)
    if n==2:
        colors = ['#66b3ff', '#fb3999']
    elif n==3:
        colors = ['#66b3ff', '#fb3999', '#ffcc99']
    elif n==4:
        colors = ['#66b3ff', '#fb3999', '#ffcc99',"#66f3ff"]
    elif n==5:
        colors = ['#66b3ff', '#fb3999', '#ffcc99',"#66f3ff",'#adcc99']
    elif n==6:
        colors = ['#66b3ff', '#fb3999', '#ffcc99',"#66f3ff",'#adcc99',"#db7f23"]

    fig1, f1 = plt.subplots()
    f1.pie(data[col].value_counts(), labels=labels, colors = colors, autopct='%
1.1f%%',shadow=False, startangle=60)
    f1.axis('equal')
    plt.tight_layout()
    plt.show()

def histo(data,col):
    plt.figure(figsize = (10, 8))
    sns.histplot(data=data, x=col, hue = data[col], fill=True)
```

Counting the % of each classe

In [12]:

```
train_data.rating.value_counts(normalize = True)
```

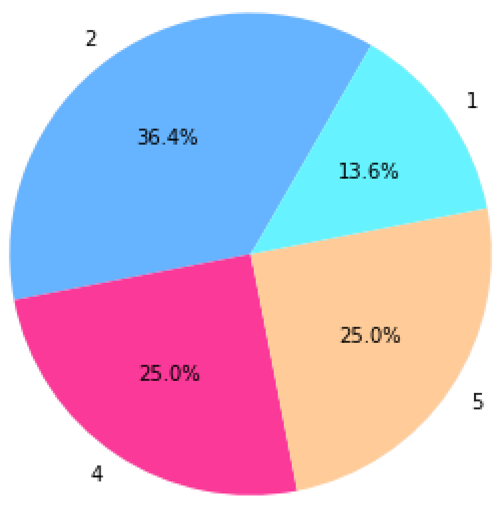
Out[12]:

```
2    0.363933
4    0.250241
5    0.249759
1    0.136067
Name: rating, dtype: float64
```

Printing the distribution of the classes

In [13]:

```
pie(train_data, "rating")
```



Repartitionning the data to 2 classes

In [15]:

```
positive_reviews = train_data[train_data["rating"] > 3]
positive_reviews["sentiment"] = "Positive"

negative_reviews = train_data[train_data["rating"] < 3]
negative_reviews["sentiment"] = "Negative"

train_data = pd.concat([positive_reviews, negative_reviews], ignore_index = True
)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pand
as-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
py

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: Sett
ingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pand
as-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
py
  """
```

preprocessing the reviews and printing the time spent & Deleting unused fields

In [16]:

```
# Cleaning Training Data
train_data[Text_Col_Train] = train_data[Text_Col_Train].apply(lambda x:   data_c
leaning(str(x)))

# Removing un-needed feilds
train_data.drop(['no','Hotel name','rating','user type','room type','nights'], a
xis = 1, inplace = True)
train_data.head(3)
```

Out[16]:

	review	sentiment
0	استثنائي سهولة إنهاء المعاملة في الاستقبال لاشيئ	Positive
1	استثنائي انصح بأختيار الاسويت و بالاخص غرفه رق...	Positive
2	جيد المكان جميل وهاديء كل شي جيد ونظيف بس كان ...	Positive

Spliting Data (Train , Evaluation)

In [42]:

```
# First setting the max_len , will be useful later for BERT Model
Extra_Len = 6 # an extra padding in length , found to be useful for increasing F
-score
Max_Len = train_data[Text_Col_Train].str.split().str.len().max() + Extra_Len
print(Max_Len)

#Spliting the Training data
Test_Size = 0.20

Rand_Seed = 42

train_set, evaluation_set = train_test_split( train_data, test_size= Test_Size,
random_state= Rand_Seed)

y=pd.get_dummies(train_data.sentiment)

train_set, X_test, y_train, y_test = train_test_split(train_data, y, test_size =
0.20, random_state = 42)

print("Train set: ")
print(train_set[Sentiment_Col_Train].value_counts())
print("---------------------------")
print ("Evaluation set: ")
print (evaluation_set[Sentiment_Col_Train].value_counts())
```

```
512
Train set:
Positive    42309
Negative    42249
Name: sentiment, dtype: int64
---------------------------
Evaluation set:
Negative    10600
Positive    10540
Name: sentiment, dtype: int64
```

Preparing BERTModel Classes

In [43]:

```
Model_Used = "UBC-NLP/MARBERT"
Task_Name = "classification"

class Dataset:
    def __init__(
        self,
        name,
        train,
        test,
        label_list,
    ):
        self.name = name
        self.train = train
        self.test = test
        self.label_list = label_list

class BERTModelDataset(Dataset):
    def __init__(self, text, target, model_name, max_len, label_map):
      super(BERTModelDataset).__init__()
      self.text = text
      self.target = target
      self.tokenizer_name = model_name
      self.tokenizer = AutoTokenizer.from_pretrained(model_name)
      self.max_len = max_len
      self.label_map = label_map

    def __len__(self):
      return len(self.text)

    def __getitem__(self,item):
      text = str(self.text[item])
      text = " ".join(text.split())

      encoded_review = self.tokenizer.encode_plus(
      text,
      max_length= self.max_len,
      add_special_tokens= True,
      return_token_type_ids=False,
      pad_to_max_length=True,
      truncation='longest_first',
      return_attention_mask=True,
      return_tensors='pt'
    )
      input_ids = encoded_review['input_ids'].to(device)
      attention_mask = encoded_review['attention_mask'].to(device)

      return InputFeatures(input_ids=input_ids.flatten(), attention_mask=attenti
on_mask.flatten(), label=self.label_map[self.target[item]])
```

Defining Needed Methods for training and evaluation

In [44]:

```
def model_init():
  return AutoModelForSequenceClassification.from_pretrained(Model_Used, return_d
ict=True, num_labels=len(label_map))

def compute_metrics(p): #p should be of type EvalPrediction
  preds = np.argmax(p.predictions, axis=1)
  assert len(preds) == len(p.label_ids)
  print(classification_report(p.label_ids,preds))
  #print(confusion_matrix(p.label_ids,preds))

  macro_f1_pos_neg = f1_score(p.label_ids,preds,average='macro',labels=[1,2])
  macro_f1 = f1_score(p.label_ids,preds,average='macro')
  macro_precision = precision_score(p.label_ids,preds,average='macro')
  macro_recall = recall_score(p.label_ids,preds,average='macro')
  acc = accuracy_score(p.label_ids,preds)
  return {
      'macro_f1' : macro_f1,
      'macro_f1_pos_neg' : macro_f1_pos_neg,
      'macro_precision': macro_precision,
      'macro_recall': macro_recall,
      'accuracy': acc
  }

def set_seed(seed):
    torch.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False
    np.random.seed(seed)
    random.seed(seed)
    os.environ['PYTHONHASHSEED'] = str(seed)
```

Build Train and Evaluation Datasets

In [45]:

```
label_list = list(train_set[Sentiment_Col_Train].unique())

print(label_list)
print(train_set[Sentiment_Col_Train].value_counts())

data_set = Dataset( "LABR", train_set, evaluation_set, label_list )

label_map = { v:index for index, v in enumerate(label_list) }
print(label_map)

train_dataset = BERTModelDataset(train_set[Text_Col_Train].to_list(),
                                 train_set[Sentiment_Col_Train].to_list(),Model_
Used,Max_Len,label_map)

evaluation_dataset = BERTModelDataset(evaluation_set[Text_Col_Train].to_list(),
                                      evaluation_set[Sentiment_Col_Train].to_lis
t(),Model_Used,Max_Len,label_map)
```

```
['Negative', 'Positive']
Positive    42309
Negative    42249
Name: sentiment, dtype: int64
{'Negative': 0, 'Positive': 1}
```

Define Training Arguments

In [46]:

```
#define training arguments
training_args = TrainingArguments("./train")
training_args.lr_scheduler_type = 'cosine'
training_args.evaluate_during_training = True
training_args.adam_epsilon =1e-8
training_args.learning_rate = 1.7825500000000001e-05 # use this with org data
training_args.fp16 = True
training_args.per_device_train_batch_size = 16
training_args.per_device_eval_batch_size = 128
training_args.gradient_accumulation_steps = 2
training_args.num_train_epochs= 2
training_args.warmup_steps = 0
training_args.evaluation_strategy = EvaluationStrategy.EPOCH
training_args.logging_steps = 200
training_args.save_steps = 100000
training_args.seed = 42
training_args.disable_tqdm = False
```

Build The Trainer

In [47]:

```
training_args.dataloader_pin_memory = False
gc.collect()
torch.cuda.empty_cache()
set_seed(Rand_Seed)

trainer = Trainer(
    model = model_init(),
    args = training_args,
    train_dataset = train_dataset,
    eval_dataset= evaluation_dataset,
    compute_metrics=compute_metrics
)

print(training_args.seed)
```

```
Some weights of the model checkpoint at UBC-NLP/MARBERT were not use
d when initializing BertForSequenceClassification: ['cls.prediction
s.bias', 'cls.predictions.transform.dense.weight', 'cls.predictions.
transform.dense.bias', 'cls.predictions.transform.LayerNorm.weight',
'cls.predictions.transform.LayerNorm.bias', 'cls.predictions.decode
r.weight', 'cls.seq_relationship.weight', 'cls.seq_relationship.bia
s']
- This IS expected if you are initializing BertForSequenceClassifica
tion from the checkpoint of a model trained on another task or with
another architecture (e.g. initializing a BertForSequenceClassificat
ion model from a BertForPreTraining model).
- This IS NOT expected if you are initializing BertForSequenceClassi
fication from the checkpoint of a model that you expect to be exactl
y identical (initializing a BertForSequenceClassification model from
a BertForSequenceClassification model).
Some weights of BertForSequenceClassification were not initialized f
rom the model checkpoint at UBC-NLP/MARBERT and are newly initialize
d: ['classifier.weight', 'classifier.bias']
You should probably TRAIN this model on a down-stream task to be abl
e to use it for predictions and inference.
```

```
42
```

Train

In [48]:

```
all_results = []

print(Max_Len)
print(training_args.learning_rate)
print(training_args.adam_epsilon)
print(training_args.warmup_steps)
trainer.train()

results = trainer.evaluate()
all_results.append(results)
print(results)
```

```
512
1.78255e-05
1e-08
0

/usr/local/lib/python3.7/dist-packages/transformers/tokenization_uti
ls_base.py:2143: FutureWarning: The `pad_to_max_length` argument is
deprecated and will be removed in a future version, use `padding=Tru
e` or `padding='longest'` to pad to the longest sequence in the batc
h, or use `padding='max_length'` to pad to a max length. In this cas
e, you can give a specific length with `max_length` (e.g. `max_lengt
h=45`) or leave max_length to None to pad to the maximal input size
of the model (e.g. 512 for Bert).
  FutureWarning,
```

[5284/5284 1:55:03, Epoch 1/2]

Epoch	Training Loss	Validation Loss	Macro F1	Macro F1 Pos Neg	Macro Precision	Macro Recall	Accuracy	Runtime
0	0.130300	0.127983	0.959319	0.479667	0.959330	0.959328	0.959319	288.315100
1	0.114000	0.142033	0.961825	0.481023	0.961943	0.961850	0.961826	287.558900

```
              precision    recall  f1-score   support

           0       0.96      0.96      0.96     10600
           1       0.96      0.96      0.96     10540

    accuracy                           0.96     21140
   macro avg       0.96      0.96      0.96     21140
weighted avg       0.96      0.96      0.96     21140


/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classificati
on.py:1580: UndefinedMetricWarning: F-score is ill-defined and being
set to 0.0 in labels with no true nor predicted samples. Use `zero_d
ivision` parameter to control this behavior.
  _warn_prf(average, "true nor predicted", "F-score is", len(true_su
m))
/usr/local/lib/python3.7/dist-packages/transformers/tokenization_uti
ls_base.py:2143: FutureWarning: The `pad_to_max_length` argument is
deprecated and will be removed in a future version, use `padding=Tru
e` or `padding='longest'` to pad to the longest sequence in the batc
h, or use `padding='max_length'` to pad to a max length. In this cas
e, you can give a specific length with `max_length` (e.g. `max_lengt
h=45`) or leave max_length to None to pad to the maximal input size
of the model (e.g. 512 for Bert).
  FutureWarning,

              precision    recall  f1-score   support

           0       0.97      0.95      0.96     10600
           1       0.95      0.97      0.96     10540

    accuracy                           0.96     21140
   macro avg       0.96      0.96      0.96     21140
weighted avg       0.96      0.96      0.96     21140
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classificati
on.py:1580: UndefinedMetricWarning: F-score is ill-defined and being
set to 0.0 in labels with no true nor predicted samples. Use `zero_d
ivision` parameter to control this behavior.
  _warn_prf(average, "true nor predicted", "F-score is", len(true_su
m))
/usr/local/lib/python3.7/dist-packages/transformers/tokenization_uti
ls_base.py:2143: FutureWarning: The `pad_to_max_length` argument is
deprecated and will be removed in a future version, use `padding=Tru
e` or `padding='longest'` to pad to the longest sequence in the batc
h, or use `padding='max_length'` to pad to a max length. In this cas
e, you can give a specific length with `max_length` (e.g. `max_lengt
h=45`) or leave max_length to None to pad to the maximal input size
of the model (e.g. 512 for Bert).
  FutureWarning,
```

[166/166 04:46]

```
              precision    recall  f1-score   support

           0       0.97      0.95      0.96     10600
           1       0.95      0.97      0.96     10540

    accuracy                           0.96     21140
   macro avg       0.96      0.96      0.96     21140
weighted avg       0.96      0.96      0.96     21140

{'eval_loss': 0.14203259348869324, 'eval_macro_f1': 0.96182463006193
84, 'eval_macro_f1_pos_neg': 0.48102337393594513, 'eval_macro_precis
ion': 0.9619432519481768, 'eval_macro_recall': 0.9618501843829437,
'eval_accuracy': 0.9618259224219489, 'eval_runtime': 287.1875, 'eval
_samples_per_second': 73.61, 'epoch': 2.0}

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classificati
on.py:1580: UndefinedMetricWarning: F-score is ill-defined and being
set to 0.0 in labels with no true nor predicted samples. Use `zero_d
ivision` parameter to control this behavior.
  _warn_prf(average, "true nor predicted", "F-score is", len(true_su
m))
```

Results

In [49]:

```
all_results
```

Out[49]:

```
[{'epoch': 2.0,
  'eval_accuracy': 0.9618259224219489,
  'eval_loss': 0.14203259348869324,
  'eval_macro_f1': 0.9618246300619384,
  'eval_macro_f1_pos_neg': 0.48102337393594513,
  'eval_macro_precision': 0.9619432519481768,
  'eval_macro_recall': 0.9618501843829437,
  'eval_runtime': 287.1875,
  'eval_samples_per_second': 73.61}]
```

In [50]:

```
from statistics import mean
mean([x['eval_macro_f1'] for x in all_results])
```

Out[50]:

```
0.9618246300619384
```