Prediction for RoBERTa after hyperparameter tuning

After experimenting with several parameter combinations that seemed plausible giving the hyperparameter tuning runs, we found a learning rate = 0.00002 and number of epochs = 13 to fine tune RoBERTa to beat the baseline.

Code

Colab Phython Setup

Connect to google drive and set the correct working directory

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

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive
```

```
import os
os.chdir('/content/gdrive/MyDrive/TxtClassComp_MattzeePrivate/')
```

```
!pwd
```

```
/content/gdrive/MyDrive/TxtClassComp_MattzeePrivate
```

Install required libraries/frameworks

```
!pip install jsonlines
!pip install pandas
```

```
Requirement already satisfied: jsonlines in /usr/local/lib/python3.6/dist-packages (1
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from js
Requirement already satisfied: pandas in /usr/local/lib/python3.6/dist-packages (1.1
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.6/dis
Requirement already satisfied: numpy>=1.15.4 in /usr/local/lib/python3.6/dist-package
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-packages (fr
```

```
!pip install simpletransformers
```

```
Requirement already satisfied: astor in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: pydeck>=0.1.dev5 in /usr/local/lib/python3.6/dist-
```

```
Requirement already satisfied: pyarrow in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: cachetools>=4.0 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: tornado>=5.0 in /usr/local/lib/python3.6/dist-pack
Requirement already satisfied: psutil>=5.0.0 in /usr/local/lib/python3.6/dist-pac
Requirement already satisfied: configparser>=3.8.1 in /usr/local/lib/python3.6/di
Requirement already satisfied: promise<3,>=2.0 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: shortuuid>=0.5.0 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: sentry-sdk>=0.4.0 in /usr/local/lib/python3.6/dist
Requirement already satisfied: docker-pycreds>=0.4.0 in /usr/local/lib/python3.6/
Requirement already satisfied: six>=1.13.0 in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: PyYAML in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: subprocess32>=3.5.3 in /usr/local/lib/python3.6/di
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-pack
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/lo
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dis
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-pack
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.6/dist-pack
Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: decorator>=3.4.0 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: entrypoints in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: jinja2 in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: jsonschema in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: toolz in /usr/local/lib/python3.6/dist-packages (f
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: gitdb<5,>=4.0.1 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: ipywidgets>=7.0.0 in /usr/local/lib/python3.6/dist
Requirement already satisfied: ipykernel>=5.1.2; python_version >= "3.4" in /usr/
Requirement already satisfied: traitlets>=4.3.2 in /usr/local/lib/python3.6/dist-

Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: smmap<4,>=3.0.1 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: nbformat>=4.2.0 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: ipython>=4.0.0; python_version >= "3.3" in /usr/lo
Requirement already satisfied: widgetsnbextension~=3.5.0 in /usr/local/lib/python
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.6/dist-pa
Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.6/dist-
Requirement already satisfied: jupyter-core in /usr/local/lib/python3.6/dist-pack
Requirement already satisfied: pygments in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: pexpect; sys_platform != "win32" in /usr/local/lib
Requirement already satisfied: pickleshare in /usr/local/lib/python3.6/dist-packa
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in /usr/local/lib/pyt
Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.6/dist
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.6/dist-package
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.6/dist-p
Requirement already satisfied: wcwidth in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: Send2Trash in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: terminado>=0.8.1 in /usr/local/lib/python3.6/dist-
Requirement already satisfied: nbconvert in /usr/local/lib/python3.6/dist-package
Requirement already satisfied: bleach in /usr/local/lib/python3.6/dist-packages (
Requirement already satisfied: testpath in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: defusedxml in /usr/local/lib/python3.6/dist-packag
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.6/d
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.6/dist
Requirement already satisfied: webencodings in /usr/local/lib/python3.6/dist-pack
```

Load libraries

```
from simpletransformers.classification import ClassificationModel,ClassificationArgs
import pandas as pd
import numpy as np
import logging
import json
import sklearn
from statistics import mean, mode
import os
#import json
import jsonlines
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import precision_recall_fscore_support
```

Load and prepare the data

```
test_file = '/content/gdrive/MyDrive/TxtClassComp_MattzeePrivate/data/test.jsonl'
train_file = '/content/gdrive/MyDrive/TxtClassComp_MattzeePrivate/data/train.jsonl'

data_train = []
iter = 1
with jsonlines.open(train_file) as f:
    for line in f.iter():
        #data = json.load(line)
        #print(line) # or whatever else you'd like to do
        #print('processing training line: ' + str(iter))
        iter +=1
        data_train.append(line)
        #data = json.loads(line)
        #print(data)


data_test = []
iter = 1
with jsonlines.open(test_file) as f:
    for line in f.iter():
        #data = json.load(line)
        #print(line) # or whatever else you'd like to do
        #print('processing test line: ' + str(iter))
        iter +=1
        data_test.append(line)
        #data = json.loads(line)
        #print(data)
print("Count of training data entries:")
print(len(data_train))
print("Count of test data entries:")
print(len(data_test))
```

```
Count of training data entries:
5000
```

```
Count of test data entries:
1800
```

```
train_data_pd = pd.DataFrame.from_dict(data_train)
test_data_pd = pd.DataFrame.from_dict(data_test)
print("Training and Test Datasets converted to Pandas DataFrames...")
```

```
Training and Test Datasets converted to Pandas DataFrames...
```

```
#!pip install nltk
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, SpaceTokenizer
from nltk.stem import PorterStemmer

#print(stopwords.words('english'))

#for j in stopwords.words('english'):
#    print(j)

#print(stopwords.words())
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
#print(train_data_pd['context'][1])
#print(train_data_pd['response'][1])
#print(train_data_pd['context'][1][0]+train_data_pd['context'][1][1])
#print(len(train_data_pd['context'][3040]))

#test = train_data_pd['context'][10][0] + train_data_pd['context'][10][1] +train_data_pd['
#print(test)
#test2 = train_data_pd['context'][:][0] + train_data_pd['context'][:][1] +train_data_pd['r
all_stopwords = stopwords.words('english')
tk = SpaceTokenizer()
ps = PorterStemmer()

for i in range(len(train_data_pd)):
    #train_data_pd['response'][i]=train_data_pd['response'][i]+train_data_pd['context'][i]
    train_data_pd['response'][i]=train_data_pd['response'][i]+train_data_pd['context'][i][
    train_data_pd['response'][i]=train_data_pd['response'][i].replace('@USER', '').strip()
    text_tokens = tk.tokenize(train_data_pd['response'][i])
    tokens_without_sw = [word for word in text_tokens if not word in all_stopwords]
    test4=""
    for i in tokens_without_sw:
        test4 = test4 + " "+ps.stem(i)
    test4.strip()
```

```
    train_data_pd['response'][i]=test4



for i in range(len(test_data_pd)):
    #test_data_pd['response'][i]=test_data_pd['response'][i]+test_data_pd['context'][i][0]
    test_data_pd['response'][i]=test_data_pd['response'][i]+test_data_pd['context'][i][1]
    test_data_pd['response'][i]=test_data_pd['response'][i].replace('@USER', '').strip().l

    text_tokens = tk.tokenize(test_data_pd['response'][i])
    tokens_without_sw = [word for word in text_tokens if not word in all_stopwords]
    test4=""
    for i in tokens_without_sw:
        test4 = test4 + " "+ps.stem(i)
    test4.strip()
    test_data_pd['response'][i]=test4



#print(train_data_pd['response'][1])
#print(len(train_data_pd['response'][1]))
print("Converted response PD data to include Context Data")
print("Converted response to lowercase and removed stop words as well as @USER")
print("Converted response to stem words using PortStemmer")
```

```
Converted response PD data to include Context Data
Converted response to lowercase and removed stop words as well as @USER
Converted response to stem words using PortStemmer
```

```
print(test_data_pd['response'][10])
```

```
define this way : 1 . desiring the good of the other ; wanting them to thrive / flour
```

```
test=train_data_pd['response'][0]+train_data_pd['context'][0][0] + train_data_pd['context'
#print(test)
test2 = test.replace('@USER', '').strip().lower()
print(test2)
test3 = test2

all_stopwords = stopwords.words('english')
#text_tokens = word_tokenize(test3)
tk = SpaceTokenizer()

text_tokens = tk.tokenize(test3)
tokens_without_sw = [word for word in text_tokens if not word in all_stopwords]
test4=""
for i in tokens_without_sw:
    test4 = test4 + " "+i
test4.strip()
```

```
#for j in stopwords.words('english'):
#    print(j)
#    test3=test3.replace(j, '')
#print(test3)
```

```
i don't get this .. obviously you do care or you would've moved right along .. instea
'get .. obviously care would've moved right along .. instead decided care troll .. c
hild named barron ... #bebest melania care less . fact . 💯a minor child deserves p
rivacy kept politics . pamela karlan , ashamed angry obviously biased public panderi
ng , using child . child named barron ... #bebest melania care less . fact . 💯'
```

```
#Define the vector of actual results:
Actual_Results = []
for  l in data_train:
    if l['label'] == 'SARCASM':
        Actual_Results.append(1)
    else:
        Actual_Results.append(0)
```

```
## Import the various SKLearn ML models for testing:

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.svm import LinearSVC
from sklearn.model_selection import cross_val_score
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
```

```
# getting training dataset features and labels
features = train_data_pd['response']
labels = train_data_pd['label']
labels = Actual_Results
```

```
# Splitting of training data into train and test data
rawdata_train, rawdata_test, rawlabels_train, rawlabels_test = train_test_split(features,

print("Training dataset split into this many train samples:")
print(len(rawdata_train))
print("Training dataset split into this many evvalidation samples:")
print(len(rawdata_test))
```

```
Training dataset split into this many train samples:
4500
```

```
Training dataset split into this many evvalidation samples:
500
```

```
train_df = pd.DataFrame({
    'text': rawdata_train.str.replace('@USER', '', regex=False).str.strip(),
    'labels': rawlabels_train
})
#rawdata_train, rawdata_test, rawlabels_train, rawlabels_test


eval_df = pd.DataFrame({
    'text': rawdata_test.str.replace('@USER', '', regex=False).str.strip(),
    'labels': rawlabels_test
})


train_df
```

	text	labels
3871	is the gig sold out ? no i'm not ... sorry . m...	0
4299	you will wait for the next hour - losing patie...	0
4719	that ' s how clueless you are . the secdef was...	0
3195	i saw a demo . great job for taking the extra ...	0
1922	yes , despite having lived in lincoln for many...	1
...	...	...
4931	you are condescendingly disrespectful and all ...	0
3264	y'all need to be able to see my likes , so you...	0
1653	yep ! i've been asked ' will they add reviews ...	1
2607	my phone is a nokia 8 so it may be an upgrade ...	0
2732	dems have been like that all my l iife . it's ...	0

4500 rows × 2 columns

```
eval_df
```

	text	labels
398	countered with #climatechange activists stuck ...	1
3833	when we share love , we get love too . our vib...	0
4836	give me nothing . just saying no one really kn...	0
4572	sinning happens nearly everywhere . as a res...	0

Training and output

```
# Create a ClassificationModel
# for training from scratch
import torch

cuda_available = torch.cuda.is_available()
print("Does system have CUDA support?")
print(cuda_available)

model = ClassificationModel('roberta', 'roberta-base', use_cuda=cuda_available) # You can
```

```
Does system have CUDA support?
True
Some weights of the model checkpoint at roberta-base were not used when initializing
- This IS expected if you are initializing RobertaForSequenceClassification from the
- This IS NOT expected if you are initializing RobertaForSequenceClassification from
Some weights of RobertaForSequenceClassification were not initialized from the model
You should probably TRAIN this model on a down-stream task to be able to use it for p
```

```
model_args = {
    "reprocess_input_data": True,
    "overwrite_output_dir": True,
    "model_args.lazy_loading" : True,
    "use_early_stopping" : True,
    "num_train_epochs" : 4,
    "learning_rate": 2.741032760877178e-05,
    #"num_train_epochs" :13,
    #"learning_rate" : 0.000020
}

# Train the model
model.train_model(train_df,args=model_args)
```

```
100%                                                4500/4500 [01:35<00:00, 47.14it/s]

Epoch 4 of 4: 100%                                  4/4 [09:09<00:00, 137.44s/it]

Epochs 0/4. Running Loss: 0.6226:                   563/563 [05:33<00:00,
100%                                                1.69it/s]

/usr/local/lib/python3.6/dist-packages/torch/optim/lr_scheduler.py:216: UserWarning:
  warnings.warn(SAVE_STATE_WARNING, UserWarning)
Epochs 1/4. Running Loss: 0.1973:                   563/563 [03:35<00:00,
```

Gen prediction

```
# getting training dataset features and labels
features_test = test_data_pd['response']
id_final_test = test_data_pd['id']

predictions_test, raw_outputs_test = model.predict(features_test)
```

```
100%                                                1800/1800 [00:01<00:00, 951.59it/s]

100%                                                225/225 [00:06<00:00, 33.83it/s]
```

```
#Writing the RoBERTa Classifier predictions to the output file: RoBERTa_answers.txt
y_pred = predictions_test
f = open("/content/gdrive/MyDrive/TxtClassComp_MattzeePrivate/RoBERTa_colab_answer.txt", "
for i in range(len(id_final_test)):
    i_result = y_pred[i]
    pred_id = id_final_test[i]
    if i_result == 1:
        f.write(pred_id + ',' + "SARCASM" +"\n")
    else:
        f.write(pred_id + ',' + "NOT_SARCASM" +"\n")
f.close()
```