

Google's strongest NLP model: BERT

<https://github.com/google-research/bert> (<https://github.com/google-research/bert>)

Rank	Model	EM	F1
	Human Performance Stanford University (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Nov 08, 2018	BERT (single model) Google AI Language	80.005	83.061
1 Nov 16, 2018	Candi-Net+BERT (single model) 42Maru NLP Team	80.106	82.862
2 Nov 09, 2018	L6Net + BERT (single model) Layer 6 AI	79.181	82.259
3 Nov 06, 2018	SLQA+BERT (single model) Alibaba DAMO NLP http://www.aclweb.org/anthology/P18-1158	77.003	80.209
4 Nov 08, 2018	BERT_base_aug (ensemble) GammaLab	76.721	79.611
5 Nov 05, 2018	MIR-MRC(F-Net) (single model) Kangwon National University, Natural Language Processing Lab. & ForceWin, KP Lab.	74.791	77.988

BERT can be used in tasks such as question answering systems, sentiment analysis, spam filtering, named entity recognition, document clustering, etc., as the infrastructure or language model for these tasks.

Importing Libraries

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

Load Data

In [2]:

```
data = pd.read_csv('Tweets.csv')
print('Dataframe:')
data.head(1)
```

Dataframe:

Out[2]:

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negati
0	570306133677760513	neutral	1.0	NaN	

In [3]:

```
# Select features
df = data[['text', 'airline_sentiment']]
print('Feature selected DataFrame:')
df.head()
```

Feature selected DataFrame:

Out[3]:

	text	airline_sentiment
0	@VirginAmerica What @dhepburn said.	neutral
1	@VirginAmerica plus you've added commercials t...	positive
2	@VirginAmerica I didn't today... Must mean I n...	neutral
3	@VirginAmerica it's really aggressive to blast...	negative
4	@VirginAmerica and it's a really big bad thing...	negative

In [4]:

```
df.shape
```

Out[4]:

(14640, 2)

Test and Train dataframes

In [5]:

```
train_df,eval_df = train_test_split(df,test_size = 0.2)
```

Building a model

In [8]:

```
#pip install simpletransformers
from simpletransformers.classification import ClassificationModel
import torch
# Create a TransformerModel
model = ClassificationModel('bert', 'bert-base-cased', num_labels=3, args={'represent_input_data': True, 'overwrite_output_dir': True}, use_cuda=False)
```

wandb: **WARNING** W&B installed but not logged in. Run `wandb login` or set the WANDB_API_KEY env variable.

Some weights of the model checkpoint at bert-base-cased were not used when initializing BertForSequenceClassification: ['cls.predictions.bias', 'cls.predictions.transform.dense.weight', 'cls.predictions.transform.dense.bias', 'cls.predictions.decoder.weight', 'cls.seq_relationship.weight', 'cls.seq_relationship.bias', 'cls.predictions.transform.LayerNorm.weight', 'cls.predictions.transform.LayerNorm.bias']

- This IS expected if you are initializing BertForSequenceClassification from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a BertForPretraining model).

- This IS NOT expected if you are initializing BertForSequenceClassification from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).

Some weights of BertForSequenceClassification were not initialized from the model checkpoint at bert-base-cased and are newly initialized: ['classifier.weight', 'classifier.bias']

You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.

In [9]:

```
def making_label(airline_sentiment):
    if(airline_sentiment=='positive'):
        return 0
    elif(airline_sentiment=='neutral'):
        return 2
    else:
        return 1

train_df['label'] = train_df['airline_sentiment'].apply(making_label)
eval_df['label'] = eval_df['airline_sentiment'].apply(making_label)
print(train_df.head())
```

```

              text  ... label
11001  @USAirways we already spoke to someone several...  ...      1
1561           @united I need help with a missing bag.  ...      1
2557  @united my flight out of BGM Cancelled Flightl...  ...      1
4148  @united here I was thinking how I could say so...  ...      1
4616  @SouthwestAir 3 hours and 80 degree difference...  ...      0
```

[5 rows x 3 columns]

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:9: SettingWithCopyWarning:
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/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
if __name__ == '__main__':
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:10: SettingWithCopyWarning:
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/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
# Remove the CWD from sys.path while we load stuff.
```

In [10]:

```
train_df2 = pd.DataFrame({
    'text': train_df['text'].replace(r'\n', ' ', regex=True),
    'label': train_df['label']
})

eval_df2 = pd.DataFrame({
    'text': eval_df['text'].replace(r'\n', ' ', regex=True),
    'label': eval_df['label']
})
```

In [11]:

```
model.train_model(train_df2)
```

```
/usr/local/lib/python3.6/dist-packages/simpletransformers/classification/classification_model.py:353: UserWarning: Dataframe headers not specified. Falling back to using column 0 as text and column 1 as labels.
```

```
"Dataframe headers not specified. Falling back to using column 0 as text and column 1 as labels."
```

```
/usr/local/lib/python3.6/dist-packages/torch/optim/lr_scheduler.py:216: UserWarning: Please also save or load the state of the optimizer when saving or loading the scheduler.
```

```
warnings.warn(SAVE_STATE_WARNING, UserWarning)
```

Out[11]:

```
(1464, 0.522169746533596)
```

```
model.train_model(train_df2)
```

```
/usr/local/lib/python3.6/dist-packages/simpletransformers/classification/classification_model.py:353: UserWarning: Dataframe headers not specified. Falling back to using column 0 as text and column 1 as labels."
```

```
100% ██████████ 11712/11712 [00:05<00:00, 1971.90it/s]
```

```
Epoch 1 of 1: 100% ██████████ 1/1 [4:21:54<00:00, 15714.03s/it]
```

```
Epochs 0/1. Running Loss: 0.3393: 100% ██████████ 1464/1464 [4:21:49<00:00, 10.73s/it]
```

```
/usr/local/lib/python3.6/dist-packages/torch/optim/lr_scheduler.py:216: UserWarning: Please also save or load the state of the optimizer when saving or loading the scheduler.
```

```
warnings.warn(SAVE_STATE_WARNING, UserWarning)
```

```
(1464, 0.522169746533596)
```

```
[ ] result, model_outputs, wrong_predictions = model.eval_model(eval_df2)
```

```
/usr/local/lib/python3.6/dist-packages/simpletransformers/classification/classification_model.py:851: UserWarning: Dataframe headers not specified. Falling back to using column 0 as text and column 1 as labels."
```

```
100% ██████████ 2928/2928 [00:01<00:00, 2007.10it/s]
```

```
Running Evaluation: 100% ██████████ 366/366 [18:57<00:00, 3.11s/it]
```

In [12]:

```
result, model_outputs, wrong_predictions = model.eval_model(eval_df2)
```

```
/usr/local/lib/python3.6/dist-packages/simpletransformers/classification/classification_model.py:851: UserWarning: Dataframe headers not specified. Falling back to using column 0 as text and column 1 as labels.
```

```
"Dataframe headers not specified. Falling back to using column 0 as text and column 1 as labels."
```

```
result, model_outputs, wrong_predictions = model.eval_model(eval_df2)
```

```
/usr/local/lib/python3.6/dist-packages/simpletransformers/classification/classification_model.py:851: UserWarning: Dataframe headers not specified. Falling back to using column 0 as text and column 1 as labels."
```

```
100% ██████████ 2928/2928 [00:01<00:00, 2007.10it/s]
```

```
Running Evaluation: 100% ██████████ 366/366 [18:57<00:00, 3.11s/it]
```

Model Evaluation

In [13]:

```
print(result)
print(model_outputs)
#print(wrong_predictions)
```

```
{'mcc': 0.6898161979010954, 'eval_loss': 0.4371291710053637}
[[-3.34957337  2.45363379 -1.35096169]
 [-2.83464003  3.16320467 -2.11451244]
 [-3.41498327  2.94220018 -1.60261059]
 ...
 [-2.8468976   3.07814193 -2.15746832]
 [-3.25913382  2.70254469 -1.55403459]
 [-2.71210432  3.05955839 -2.21712756]]
```

In [14]:

```
lst = []
for arr in model_outputs:
    lst.append(np.argmax(arr))
```

In [15]:

```
true = eval_df2['label'].tolist()
predicted = lst
```

In [16]:

```
import sklearn
mat = sklearn.metrics.confusion_matrix(true , predicted)
mat
```

Out[16]:

```
array([[ 345,   84,   45],
       [  33, 1687,   83],
       [  50,  186,  415]])
```

In [17]:

```
print(sklearn.metrics.classification_report(true,predicted,target_names=[ 'positive', 'neutral', 'negative' ]))
```

	precision	recall	f1-score	support
positive	0.81	0.73	0.76	474
neutral	0.86	0.94	0.90	1803
negative	0.76	0.64	0.70	651
accuracy			0.84	2928
macro avg	0.81	0.77	0.79	2928
weighted avg	0.83	0.84	0.83	2928

Our model has an accuracy rate of 83.5% !

In [18]:

```
sklearn.metrics.accuracy_score(true,predicted)
```

Out[18]:

0.835724043715847

Test statement

In [19]:

```
def get_result(text):
    result = model.predict([text])
    pos = np.where(result[1][0] == np.amax(result[1][0]))
    pos = int(pos[0])
    sentiment_dict = {0:'positive',1:'negative',2:'neutral'}
    print(sentiment_dict[pos])
    return
```

In [20]:

```
## positive statement
get_result("You are so nice.")
```

positive

In [21]:

```
## negative statement
get_result('I hate you.')
```

negative

Save model

In [22]:

```
from sklearn.externals import joblib
# Save model
joblib.dump(model, 'ML-Model.pkl')
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/externals/joblib/__in
it__.py:15: FutureWarning: sklearn.externals.joblib is deprecated in
0.21 and will be removed in 0.23. Please import this functionality d
irectly from joblib, which can be installed with: pip install jobli
b. If this warning is raised when loading pickled models, you may ne
ed to re-serialize those models with scikit-learn 0.21+.
    warnings.warn(msg, category=FutureWarning)
```

Out[22]:

['ML-Model.pkl']