

Model evaluation

July 10, 2019

```
[3]: import numpy as np
import pandas as pd

import os
print(os.listdir("."))
```

```
['.ipynb_checkpoints', 'Model evaluation BERT-ABSA v1.ipynb', 'test_ep_6.txt',
'test_NLI_B.tsv']
```

```
[4]: test_model = pd.read_csv('test_ep_6.txt', sep=' ', header=None)
test_model.head()
```

```
[4]:   0      1      2
0  1  0.000016  0.999984
1  0  0.999978  0.000022
2  0  0.999992  0.000008
3  1  0.000019  0.999981
4  0  0.999977  0.000023
```

```
[5]: test_orig = pd.read_csv('test_NLI_B.tsv', sep='\t')
test_orig.head()
```

```
[5]:   id      sentence1  polarity  target \
0  0  if you want to go out drinking where the cool ...      None  location1
1  0  if you want to go out drinking where the cool ...  Positive  location1
2  0  if you want to go out drinking where the cool ...  Negative  location1
3  0  if you want to go out drinking where the cool ...      None  location1
4  0  if you want to go out drinking where the cool ...  Positive  location1

   aspect  label
0  general      1
1  general      0
2  general      0
3   price      1
4   price      0
```

```
[6]: test = pd.concat([test_model, test_orig], axis = 1)
test['label_pred'] = test[0]
test.head()
```

```
[6]:   0      1      2 id \
0  1  0.000016  0.999984  0
1  0  0.999978  0.000022  0
2  0  0.999992  0.000008  0
3  1  0.000019  0.999981  0
4  0  0.999977  0.000023  0
```

```

                                sentence1  polarity    target \
0  if you want to go out drinking where the cool ...      None  location1
1  if you want to go out drinking where the cool ...  Positive  location1
2  if you want to go out drinking where the cool ...  Negative  location1
3  if you want to go out drinking where the cool ...      None  location1
4  if you want to go out drinking where the cool ...  Positive  location1
```

```

    aspect  label  label_pred
0  general      1          1
1  general      0          0
2  general      0          0
3   price      1          1
4   price      0          0
```

```
[7]: test_pivot = test.pivot_table(index=['sentence1', 'target', 'aspect'],
    ↪columns='polarity', values=['label', 'label_pred'], aggfunc=sum).fillna(0)
test_pivot.head(12)
```

```
[7]: label \
polarity
Negative
sentence1
# location1 arts centre 0 , free      target    aspect
location1 general
0
                                price
0
                                safety
0
                                transit location
0
' punk rock mentality , location1 is the epice... location1 general
0
                                price
0
                                safety
0
                                transit location
0
' ave a bit of chow at harrods , then go to the... location1 general
0
                                price
```

0			
			safety
0			
			transit location
0			
	\		
polarity			
None			
sentence1		target	aspect
# location1 arts centre 0 , free		location1	general
1			
			price
1			
			safety
1			
			transit location
1			
' punk rock mentality , location1 is the epice... location1			general
1			
			price
1			
			safety
1			
			transit location
1			
' ave a bit of chow at harrods , then go to the... location1			general
1			
			price
1			
			safety
1			
			transit location
1			
	\		
polarity			
Positive			
sentence1		target	aspect
# location1 arts centre 0 , free		location1	general
0			
			price
0			
			safety
0			
			transit location
0			

```

' punk rock mentality , location1 is the epice... location1 general
0
                                price
0
                                safety
0
                                transit location
0
' ave a bit of chow at harrods , then go to the... location1 general
0
                                price
0
                                safety
0
                                transit location
0

label_pred \
polarity
Negative
sentence1
# location1 arts centre 0 , free
0
                                target    aspect
                                location1 general
0
                                price
0
                                safety
0
                                transit location
0
' punk rock mentality , location1 is the epice... location1 general
0
                                price
0
                                safety
0
                                transit location
0
' ave a bit of chow at harrods , then go to the... location1 general
0
                                price
0
                                safety
0
                                transit location
0

\

```

polarity		target	aspect
None			
sentence1		location1	general
# location1 arts centre 0 , free			
1			price
1			safety
1			transit location
1			
' punk rock mentality , location1 is the epice... location1		general	
1			price
1			safety
1			transit location
1			
' ave a bit of chow at harrods , then go to the... location1		general	
1			price
1			safety
1			transit location
1			

polarity		target	aspect
Positive			
sentence1		location1	general
# location1 arts centre 0 , free			
0			price
0			safety
0			transit location
0			
' punk rock mentality , location1 is the epice... location1		general	
0			price
0			safety
0			transit location

```

0
' ave a bit of chow at harrods , then go to the... location1 general
0
                                price
0
                                safety
0
                                transit location
0

```

```

[8]: test_pivot[test_pivot['label']['Negative'] > 1].size
+ test_pivot[test_pivot['label']['Positive'] > 1].size,
+ test_pivot[test_pivot['label']['None'] > 1].size

# sent1 + target + aspecto + polarity que se repitieron

```

[8]: 186

```

[13]: test_pivot['polarity'] = np.select([test_pivot['label']['Positive'] >= 1,
                                         test_pivot['label']['Negative'] >= 1,
                                         test_pivot['label']['None'] >= 1], ['P', 'N', '-'])
test_pivot['polarity_pred'] = np.select([test_pivot['label_pred']['Positive'] >= 1,
                                         test_pivot['label_pred']['Negative'] >= 1,
                                         test_pivot['label_pred']['None'] >= 1], ['P', 'N', '-'])
test_pivot.head()

```

```

[13]: label \
polarity
Negative
sentence1          target    aspect
# location1 arts centre 0 , free    location1 general
0
                                price
0
                                safety
0
                                transit location
0
' punk rock mentality , location1 is the epice... location1 general
0

\
polarity
None
sentence1          target    aspect

```

```

# location1 arts centre 0 , free          location1 general
1
                                         price
1
                                         safety
1
                                         transit location
1
' punk rock mentality , location1 is the epice... location1 general
1

```

```

\
polarity
Positive
sentence1
# location1 arts centre 0 , free          target    aspect
0                                           location1 general
                                         price
0
                                         safety
0
                                         transit location
0
' punk rock mentality , location1 is the epice... location1 general
0

```

```

label_pred \
polarity
Negative
sentence1
# location1 arts centre 0 , free          target    aspect
0                                           location1 general
                                         price
0
                                         safety
0
                                         transit location
0
' punk rock mentality , location1 is the epice... location1 general
0

```

```

\
polarity
None
sentence1
# location1 arts centre 0 , free          target    aspect
1                                           location1 general

```

1		price
		safety
1		transit location
1		
'	punk rock mentality , location1 is the epice... location1	general
1		

\

polarity		
Positive		
sentence1	target	aspect
# location1 arts centre 0 , free	location1	general
0		price
0		safety
0		transit location
0		
'	punk rock mentality , location1 is the epice... location1	general
0		

polarity \		
polarity		
sentence1	target	aspect
# location1 arts centre 0 , free	location1	general
-		price
-		safety
-		transit location
-		
'	punk rock mentality , location1 is the epice... location1	general
-		

polarity_pred		
polarity		
sentence1	target	aspect
# location1 arts centre 0 , free	location1	general
-		price
-		safety
-		

transit location

```
-  
' punk rock mentality , location1 is the epice... location1 general  
-
```

```
[14]: from sklearn.metrics import confusion_matrix  
from sklearn.utils.multiclass import unique_labels  
import matplotlib.pyplot as plt  
from matplotlib.pyplot import figure  
import math  
from matplotlib.pyplot import figure  
import seaborn as sns  
sns.set(style='darkgrid')  
  
def plot_confusion_matrix(y_true, y_pred, classes, normalize=True, title=None,  
    cmap=plt.cm.Blues):  
    if not title:  
        title = 'Normalized confusion matrix' if normalize else 'Confusion  
    matrix, without normalization'  
  
    cm = confusion_matrix(y_true, y_pred)  
    cm_norm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis] * 100  
  
    fig, ax = plt.subplots(figsize = (5, 4), dpi=150)  
  
    im = ax.imshow(cm_norm, interpolation='nearest', cmap=cmap)  
    ax.figure.colorbar(im, ax=ax)  
    ax.grid(False)  
  
    ax.set(xticks=np.arange(cm.shape[1]),  
          yticks=np.arange(cm.shape[0]),  
          xticklabels=classes,  
          yticklabels=classes,  
          title=title,  
          ylabel='True label',  
          xlabel='Predicted label')  
  
    plt.setp(ax.get_xticklabels(), rotation=45, ha="right",  
    rotation_mode="anchor")  
  
    fmt = 'd'  
    fmt_norm = '.2f'  
  
    thresh = 50  
  
    for i in range(cm.shape[0]):  
        for j in range(cm.shape[1]):
```

```

    if cm[i, j] == 0:
        continue
    ax.text(j, i, '\n' + format(cm[i, j], fmt), fontsize=8,
            ha="center", va="top",
            color="white" if cm_norm[i, j] > thresh else "black")

    if not math.isnan(cm_norm[i, j]):
        ax.text(j, i, format(cm_norm[i, j], fmt_norm) + '%', fontsize=8,
            ha="center", va="bottom",
            color="white" if cm_norm[i, j] > thresh else "black")

fig.tight_layout()
return ax

```

```

[15]: y_real = test_pivot['polarity'].values
      y_pred = test_pivot['polarity_pred'].values

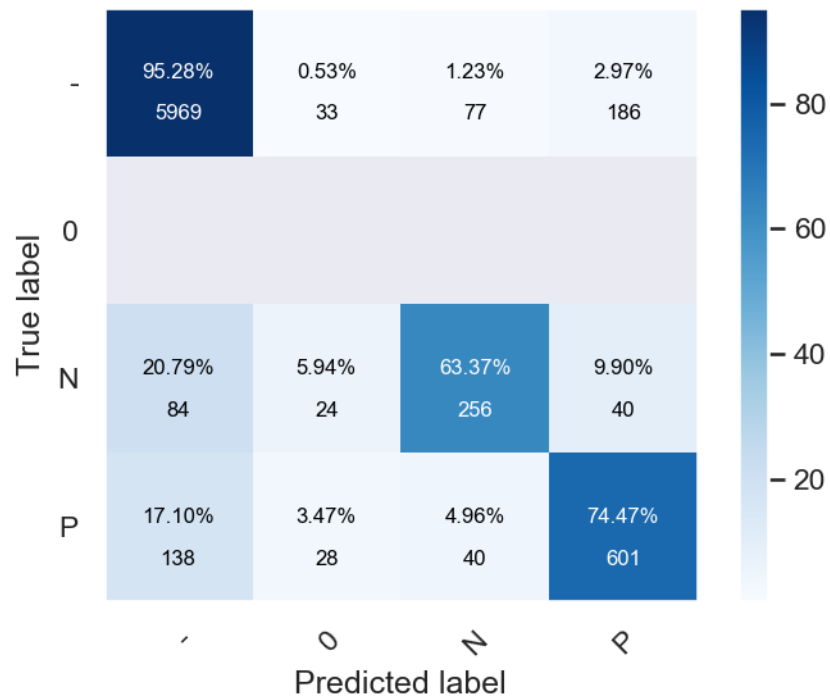
      plot_confusion_matrix(y_real, y_pred, classes=unique_labels(y_pred))

```

C:\Users\Hugo\Anaconda3\envs\pytorch_Py37\lib\site-packages\ipykernel_launcher.py:15: RuntimeWarning: invalid value encountered in true_divide

```
from ipykernel import kernelapp as app
```

[15]: <matplotlib.axes._subplots.AxesSubplot at 0x16bc80d15c0>



```
[16]: from sklearn.metrics import roc_auc_score

P_real, P_pred = [1 if x == 'P' else 0 for x in y_real], [1 if x == 'P' else 0
    →for x in y_pred]
N_real, N_pred = [1 if x == 'N' else 0 for x in y_real], [1 if x == 'N' else 0
    →for x in y_pred]
Non_real, Non_pred = [1 if x == '-' else 0 for x in y_real], [1 if x == '-'
    →else 0 for x in y_pred]

roc_auc_P = roc_auc_score(P_real, P_pred)
roc_auc_N = roc_auc_score(N_real, N_pred)
roc_auc_Non = roc_auc_score(Non_real, Non_pred)

#print('roc_auc score\n P = {0}\n N = {1}\n - = {2}'.format(roc_auc_P,
    →roc_auc_N, roc_auc_Non))
```

```
[17]: from sklearn.metrics import roc_curve

def plot_roc_curve(fpr, tpr, label=None):
    plt.plot(fpr, tpr, linewidth = 2, label=label)
    plt.plot([0, 1], [0, 1], 'k--')
    plt.axis([0, 1, 0, 1])
    plt.xlabel('False Positive Rate \n(1 - Especificity)')
    plt.ylabel('True Positive Rate \n(Sensibility - Recall)')
    plt.legend()

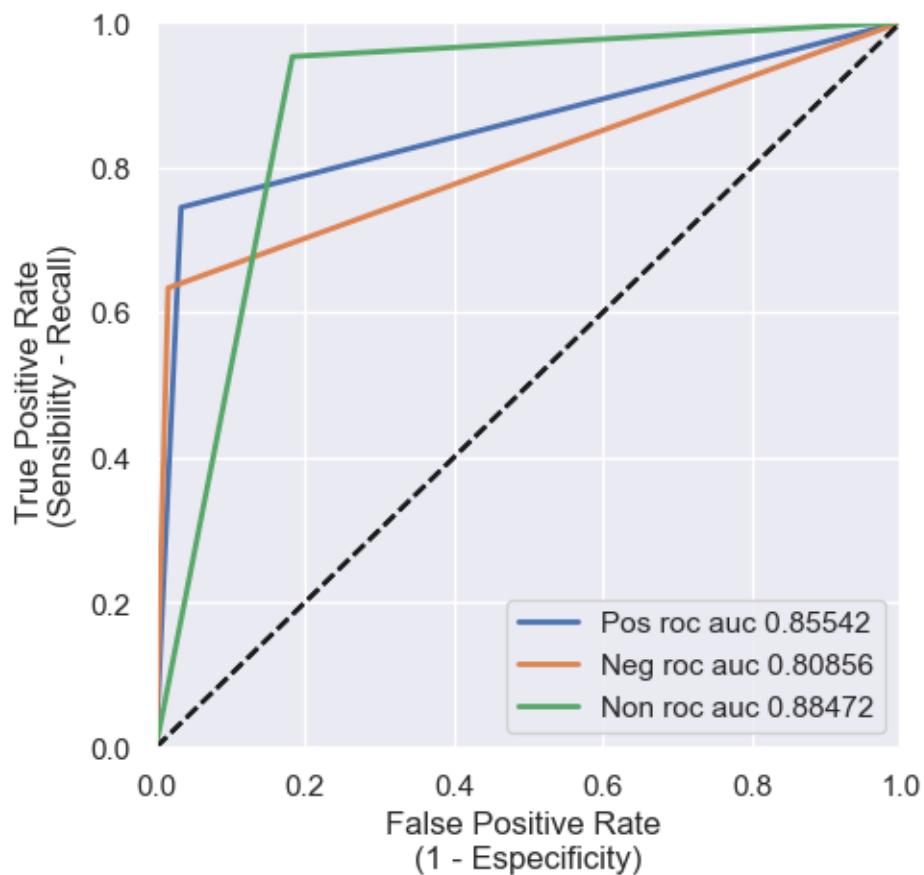
fig, ax = plt.subplots(figsize = (5, 5), dpi=100)

fpr, tpr, thresholds = roc_curve(P_real, P_pred)
plot_roc_curve(fpr, tpr, 'Pos roc auc {0:.5f}'.format(roc_auc_P))

fpr, tpr, thresholds = roc_curve(N_real, N_pred)
plot_roc_curve(fpr, tpr, 'Neg roc auc {0:.5f}'.format(roc_auc_N))

fpr, tpr, thresholds = roc_curve(Non_real, Non_pred)
plot_roc_curve(fpr, tpr, 'Non roc auc {0:.5f}'.format(roc_auc_Non))

plt.show()
```



```
[234]: test_asp_pol = test_pivot.copy().reset_index()
```

```
[281]: test_asp_pol.head()
```

```
[281]:
```

	sentence1	target	\
polarity			
0	# location1 arts centre 0 , free	location1	
1	# location1 arts centre 0 , free	location1	
2	# location1 arts centre 0 , free	location1	
3	# location1 arts centre 0 , free	location1	
4	' punk rock mentality , location1 is the epic...	location1	

	aspect	label	label_pred	\			
		Negative	None	Positive	Negative	None	Positive
polarity							
0	general	0	1	0	0	1	0
1	price	0	1	0	0	1	0
2	safety	0	1	0	0	1	0
3	transit location	0	1	0	0	1	0
4	general	0	1	0	0	1	0

polarity	polarity_pred	asp_pol_real	asp_pol_pred
----------	---------------	--------------	--------------

polarity				
0	-	-	G-	G-
1	-	-	\$-	\$-
2	-	-	S-	S-
3	-	-	L-	L-
4	-	-	G-	G-

```
[274]: test_asp_pol['asp_pol_real'] = np.select([(test_asp_pol['aspect'] == 'general') &
→& (test_asp_pol['label']['Positive'] >= 1),
                                                (test_asp_pol['aspect'] == 'general') &
→& (test_asp_pol['label']['Negative'] >= 1),
                                                (test_asp_pol['aspect'] == 'general') &
→& (test_asp_pol['label']['None'] >= 1),
                                                (test_asp_pol['aspect'] == 'price') &
→(test_asp_pol['label']['Positive'] >= 1),
                                                (test_asp_pol['aspect'] == 'price') &
→(test_asp_pol['label']['Negative'] >= 1),
                                                (test_asp_pol['aspect'] == 'price') &
→(test_asp_pol['label']['None'] >= 1),
                                                (test_asp_pol['aspect'] == 'transit_
→location') & (test_asp_pol['label']['Positive'] >= 1),
                                                (test_asp_pol['aspect'] == 'transit_
→location') & (test_asp_pol['label']['Negative'] >= 1),
                                                (test_asp_pol['aspect'] == 'transit_
→location') & (test_asp_pol['label']['None'] >= 1),
                                                (test_asp_pol['aspect'] == 'safety') &
→& (test_asp_pol['label']['Positive'] >= 1),
                                                (test_asp_pol['aspect'] == 'safety') &
→& (test_asp_pol['label']['Negative'] >= 1),
                                                (test_asp_pol['aspect'] == 'safety') &
→& (test_asp_pol['label']['None'] >= 1)],
                                                ['GP', 'GN', 'G-', '$P', '$N', '$-',
→'LP', 'LN', 'L-', 'SP', 'SN', 'S-'])

test_asp_pol['asp_pol_pred'] = np.select([(test_asp_pol['aspect'] == 'general') &
→& (test_asp_pol['label_pred']['Positive'] >= 1),
                                                (test_asp_pol['aspect'] == 'general') &
→& (test_asp_pol['label_pred']['Negative'] >= 1),
                                                (test_asp_pol['aspect'] == 'general') &
→& (test_asp_pol['label_pred']['None'] >= 1),
                                                (test_asp_pol['aspect'] == 'price') &
→(test_asp_pol['label_pred']['Positive'] >= 1),
                                                (test_asp_pol['aspect'] == 'price') &
→(test_asp_pol['label_pred']['Negative'] >= 1),
                                                (test_asp_pol['aspect'] == 'price') &
→(test_asp_pol['label_pred']['None'] >= 1),
```

```

        (test_asp_pol['aspect'] == 'transit_
→location') & (test_asp_pol['label_pred']['Positive'] >= 1),
        (test_asp_pol['aspect'] == 'transit_
→location') & (test_asp_pol['label_pred']['Negative'] >= 1),
        (test_asp_pol['aspect'] == 'transit_
→location') & (test_asp_pol['label_pred']['None'] >= 1),
        (test_asp_pol['aspect'] == 'safety')_
→& (test_asp_pol['label_pred']['Positive'] >= 1),
        (test_asp_pol['aspect'] == 'safety')_
→& (test_asp_pol['label_pred']['Negative'] >= 1),
        (test_asp_pol['aspect'] == 'safety')_
→& (test_asp_pol['label_pred']['None'] >= 1)],
        ['GP', 'GN', 'G-', '$P', '$N', '$-',_
→'LP', 'LN', 'L-', 'SP', 'SN', 'S-'])

```

[282]: test_asp_pol.head()

```

[282]:
                                sentence1      target  \
polarity
0                # location1 arts centre 0 , free  location1
1                # location1 arts centre 0 , free  location1
2                # location1 arts centre 0 , free  location1
3                # location1 arts centre 0 , free  location1
4      ' punk rock mentality , location1 is the epic...  location1

                                aspect      label      label_pred      \
polarity                                Negative None Positive  Negative None Positive
0                general                0      1      0                0      1      0
1                price                  0      1      0                0      1      0
2                safety                  0      1      0                0      1      0
3      transit location                  0      1      0                0      1      0
4                general                0      1      0                0      1      0

                                polarity polarity_pred asp_pol_real asp_pol_pred
polarity
0                -                -                G-                G-
1                -                -                $-                $-
2                -                -                S-                S-
3                -                -                L-                L-
4                -                -                G-                G-

```

```

[294]: y_real = test_asp_pol['asp_pol_real'].values
y_pred = test_asp_pol['asp_pol_pred'].values

plot_confusion_matrix(y_real, y_pred, classes=unique_labels(y_pred))

```

C:\Users\Hugo\Anaconda3\envs\pytorch_Py37\lib\site-packages\ipykernel_launcher.py:15: RuntimeWarning: invalid value encountered in

```

true_divide
from ipykernel import kernelapp as app

```

[294]: <matplotlib.axes._subplots.AxesSubplot at 0x1d81d81c828>

