Model Evaluation 190717

July 17, 2019

[1]:

```
import numpy as np
import pandas as pd

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

```
['.ipynb_checkpoints', 'dev_NLI_B.tsv', 'Model Evaluation 190717.ipynb',
'test_ep_1.txt', 'test_ep_2.txt', 'test_ep_3.txt', 'test_ep_4.txt']
```

[2]:

```
test_orig = pd.read_csv('dev_NLI_B.tsv', sep='\t')
test_orig.head()
```

[2]:

```
     id                           sentence1  polarity  \
0  1262   Tienda de Autoservicio. Siempre bien  Positive
1  1262  Tienda de Autoservicio. Siempre bien.  Positive
2  1262   Tienda de Autoservicio. Siempre bien      None
3  1262  Tienda de Autoservicio. Siempre bien.      None
4  1262   Tienda de Autoservicio. Siempre bien  Negative

                context   target   aspect  label
0  Tienda de Autoservicio  general  general      1
1  Tienda de Autoservicio  general  general      1
2  Tienda de Autoservicio  general  general      0
3  Tienda de Autoservicio  general  general      0
4  Tienda de Autoservicio  general  general      0
```

[3]:

```
from glob import glob

test_models = [pd.read_csv(f, sep=' ', header=None, usecols=[0]) for f in␣
 ↪glob('test_ep_*.txt')]
for i, t in enumerate(test_models):
    t.columns = ['label_pred_{0}'.format(i)]

test_model = pd.concat(test_models, axis = 1)
test_model.head()
```

[3]:

```
   label_pred_0  label_pred_1  label_pred_2  label_pred_3
0             1             1             1             1
```

```
1              1              1              1              1
2              0              0              0              0
3              0              0              0              0
4              0              0              0              0
```

[4]:
```
test = pd.concat([test_model, test_orig], axis = 1)
test.head()
```

[4]:
```
   label_pred_0  label_pred_1  label_pred_2  label_pred_3    id  \
0             1             1             1             1  1262
1             1             1             1             1  1262
2             0             0             0             0  1262
3             0             0             0             0  1262
4             0             0             0             0  1262

                           sentence1  polarity                 context  \
0   Tienda de Autoservicio. Siempre bien  Positive  Tienda de Autoservicio
1  Tienda de Autoservicio. Siempre bien.  Positive  Tienda de Autoservicio
2   Tienda de Autoservicio. Siempre bien      None  Tienda de Autoservicio
3  Tienda de Autoservicio. Siempre bien.      None  Tienda de Autoservicio
4   Tienda de Autoservicio. Siempre bien  Negative  Tienda de Autoservicio

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

[5]:
```
test['y_real'] = np.select([(test['aspect'] == 'general') & (test['polarity']␣
 ↪== 'Positive') & (test['label'] == 0),
                                (test['aspect'] == 'general') &␣
 ↪(test['polarity'] == 'Negative') & (test['label'] == 0),
                                (test['aspect'] == 'general') &␣
 ↪(test['polarity'] == 'None') & (test['label'] == 0),
                                (test['aspect'] == 'servicio') &␣
 ↪(test['polarity'] == 'Positive') & (test['label'] == 0),
                                (test['aspect'] == 'servicio') &␣
 ↪(test['polarity'] == 'Negative') & (test['label'] == 0),
                                (test['aspect'] == 'servicio') &␣
 ↪(test['polarity'] == 'None') & (test['label'] == 0),
                                (test['aspect'] == 'ambiente') &␣
 ↪(test['polarity'] == 'Positive') & (test['label'] == 0),
                                (test['aspect'] == 'ambiente') &␣
 ↪(test['polarity'] == 'Negative') & (test['label'] == 0),
                                (test['aspect'] == 'ambiente') &␣
 ↪(test['polarity'] == 'None') & (test['label'] == 0),
                                 (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'Positive') & (test['label'] == 0),
                                 (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'Negative') & (test['label'] == 0),
                                 (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'None') & (test['label'] == 0),
                                 (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'Positive') & (test['label'] == 0),
                                 (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'Negative') & (test['label'] == 0),
                                 (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'None') & (test['label'] == 0),
                                 (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'Positive') & (test['label'] == 0),
                                 (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'Negative') & (test['label'] == 0),
                                 (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'None') & (test['label'] == 0),

                                 (test['aspect'] == 'general') &␣
↪(test['polarity'] == 'Positive') & (test['label'] == 1),
                                 (test['aspect'] == 'general') &␣
↪(test['polarity'] == 'Negative') & (test['label'] == 1),
                                 (test['aspect'] == 'general') &␣
↪(test['polarity'] == 'None') & (test['label'] == 1),
                                 (test['aspect'] == 'servicio') &␣
↪(test['polarity'] == 'Positive') & (test['label'] == 1),
                                 (test['aspect'] == 'servicio') &␣
↪(test['polarity'] == 'Negative') & (test['label'] == 1),
                                 (test['aspect'] == 'servicio') &␣
↪(test['polarity'] == 'None') & (test['label'] == 1),
                                 (test['aspect'] == 'ambiente') &␣
↪(test['polarity'] == 'Positive') & (test['label'] == 1),
                                 (test['aspect'] == 'ambiente') &␣
↪(test['polarity'] == 'Negative') & (test['label'] == 1),
                                 (test['aspect'] == 'ambiente') &␣
↪(test['polarity'] == 'None') & (test['label'] == 1),
                                 (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'Positive') & (test['label'] == 1),
                                 (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'Negative') & (test['label'] == 1),
                                 (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'None') & (test['label'] == 1),
                                 (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'Positive') & (test['label'] == 1),
                                  (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'Negative') & (test['label'] == 1),
                                  (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'None') & (test['label'] == 1),
                                  (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'Positive') & (test['label'] == 1),
                                  (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'Negative') & (test['label'] == 1),
                                  (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'None') & (test['label'] == 1),
                                ],
                                        ['GP0', 'GN0', 'G-0',
                                         'SP0', 'SN0', 'S-0',
                                         'AP0', 'AN0', 'A-0',
                                         '$P0', '$N0', '$-0',
                                         'CP0', 'CN0', 'C-0',
                                         'UP0', 'UN0', 'U-0',

                                         'GP1', 'GN1', 'G-1',
                                         'SP1', 'SN1', 'S-1',
                                         'AP1', 'AN1', 'A-1',
                                         '$P1', '$N1', '$-1',
                                         'CP1', 'CN1', 'C-1',
                                         'UP1', 'UN1', 'U-1'])
```

[6]:

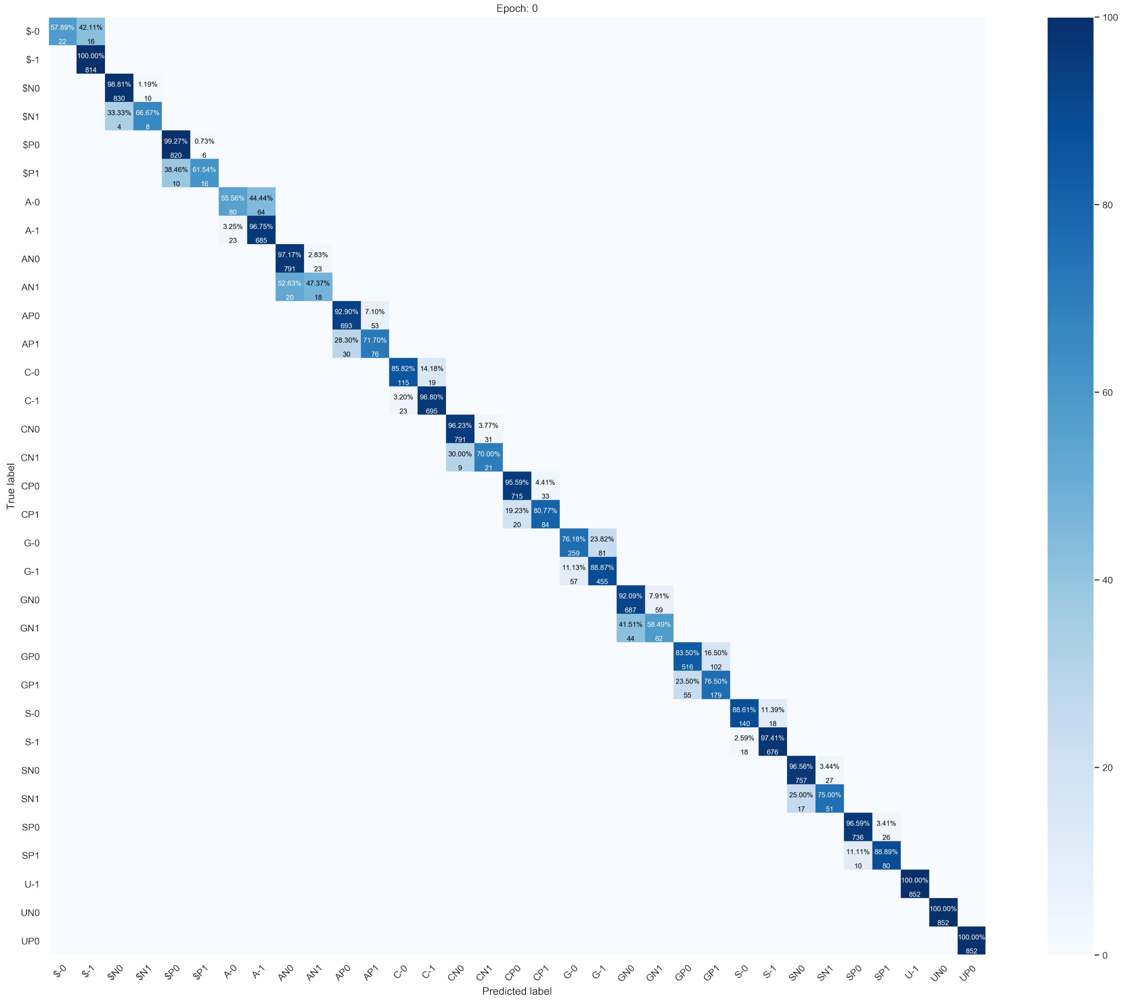
```
for k in test.keys():
    if 'label_pred_' in k:
        test['y_' + k] = np.select([(test['aspect'] == 'general') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 0),
                                  (test['aspect'] == 'general') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 0),
                                  (test['aspect'] == 'general') &␣
↪(test['polarity'] == 'None') & (test[k] == 0),
                                  (test['aspect'] == 'servicio') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 0),
                                  (test['aspect'] == 'servicio') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 0),
                                  (test['aspect'] == 'servicio') &␣
↪(test['polarity'] == 'None') & (test[k] == 0),
                                  (test['aspect'] == 'ambiente') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 0),
                                  (test['aspect'] == 'ambiente') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 0),
                                  (test['aspect'] == 'ambiente') &␣
↪(test['polarity'] == 'None') & (test[k] == 0),
                (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 0),
                (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 0),
                (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'None') & (test[k] == 0),
                (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 0),
                (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 0),
                (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'None') & (test[k] == 0),
                (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 0),
                (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 0),
                (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'None') & (test[k] == 0),

                (test['aspect'] == 'general') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 1),
                (test['aspect'] == 'general') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 1),
                (test['aspect'] == 'general') &␣
↪(test['polarity'] == 'None') & (test[k] == 1),
                (test['aspect'] == 'servicio') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 1),
                (test['aspect'] == 'servicio') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 1),
                (test['aspect'] == 'servicio') &␣
↪(test['polarity'] == 'None') & (test[k] == 1),
                (test['aspect'] == 'ambiente') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 1),
                (test['aspect'] == 'ambiente') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 1),
                (test['aspect'] == 'ambiente') &␣
↪(test['polarity'] == 'None') & (test[k] == 1),
                (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 1),
                (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 1),
                (test['aspect'] == 'precio') &␣
↪(test['polarity'] == 'None') & (test[k] == 1),
                (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 1),
                                  (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 1),
                                  (test['aspect'] == 'comida') &␣
↪(test['polarity'] == 'None') & (test[k] == 1),
                                  (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'Positive') & (test[k] == 1),
                                  (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'Negative') & (test[k] == 1),
                                  (test['aspect'] == 'ubicaciÃn') &␣
↪(test['polarity'] == 'None') & (test[k] == 1),
                                 ],
                                        ['GP0', 'GN0', 'G-0',
                                         'SP0', 'SN0', 'S-0',
                                         'AP0', 'AN0', 'A-0',
                                         '$P0', '$N0', '$-0',
                                         'CP0', 'CN0', 'C-0',
                                         'UP0', 'UN0', 'U-0',

                                         'GP1', 'GN1', 'G-1',
                                         'SP1', 'SN1', 'S-1',
                                         'AP1', 'AN1', 'A-1',
                                         '$P1', '$N1', '$-1',
                                         'CP1', 'CN1', 'C-1',
                                         'UP1', 'UN1', 'U-1'])
```

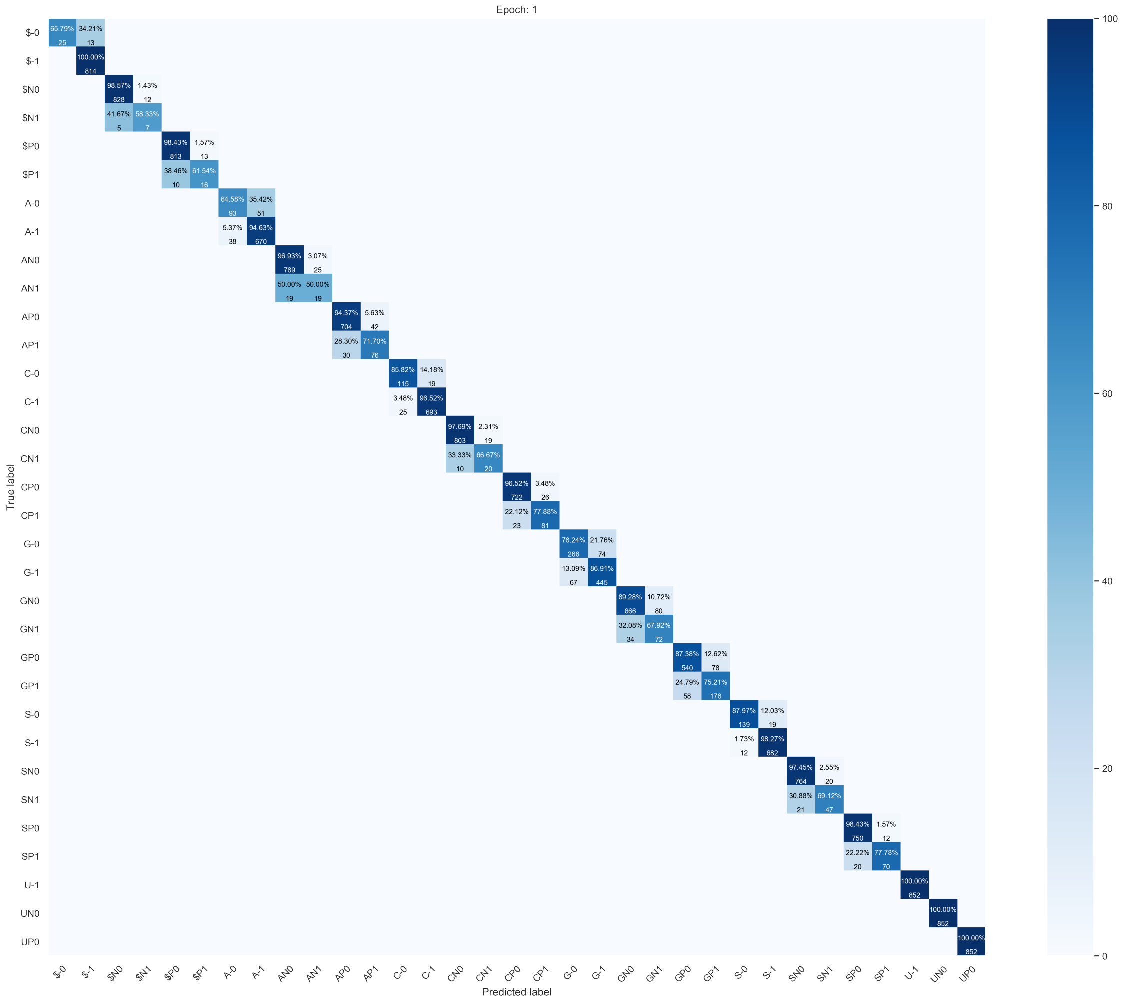
[7]:

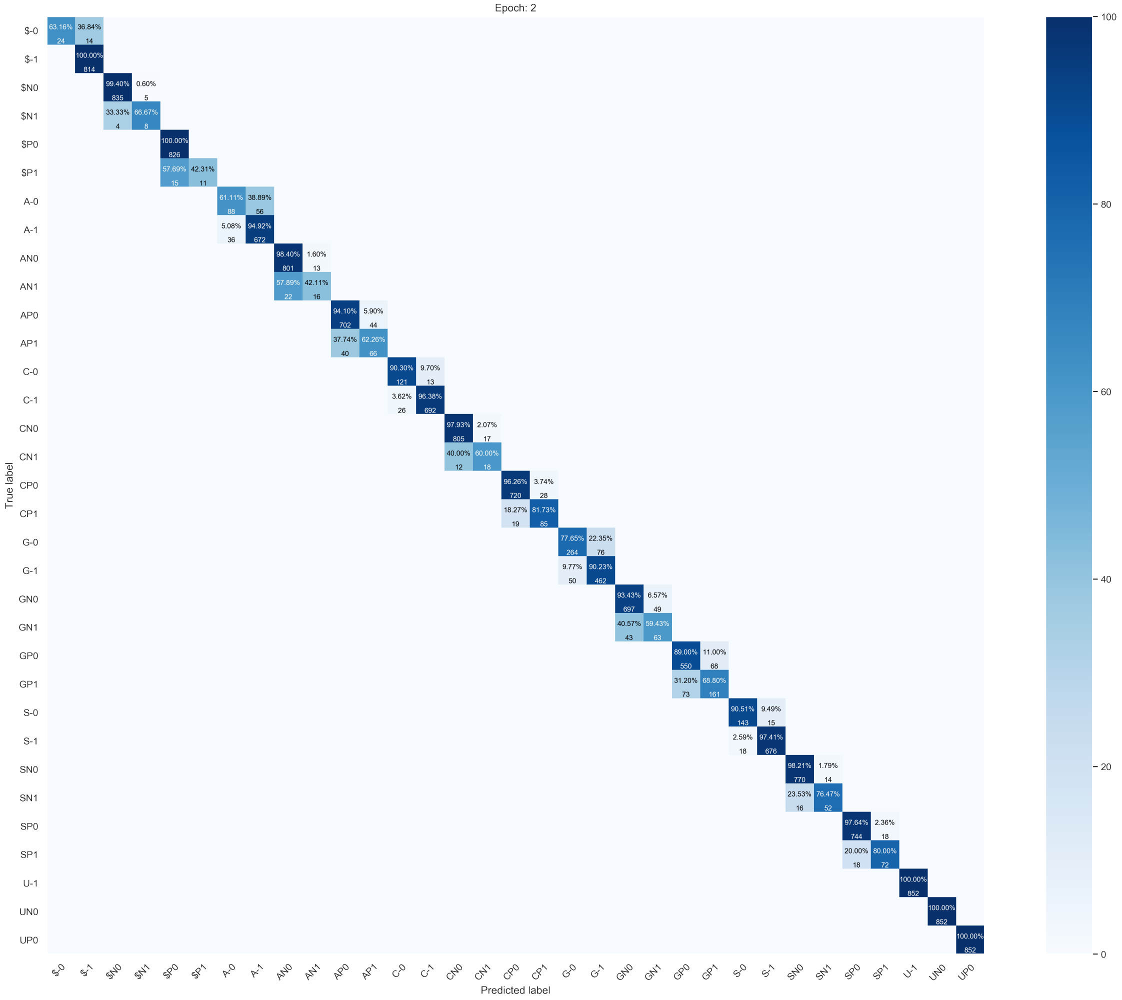
```
from sklearn.metrics import confusion_matrix
from sklearn.utils.multiclass import unique_labels
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
import math
from matplotlib.pyplot import figure
import seaborn as sns
sns.set(style='darkgrid')

def plot_confusion_matrix(y_true, y_pred, classes, title="", cmap=plt.cm.Blues,␣
↪clean=False, figsize=(20, 16), dpi=300, showLabels=True):

    cm = confusion_matrix(y_true, y_pred)
    cm_norm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis] * 100

    if clean:
        # indexes of No's '0'  and  None's '-'
        indexes = [i for i, c in enumerate(classes) if c.endswith('0') or  '-'␣
↪in c]

        cm = np.delete(cm, indexes, axis=0)
        cm = np.delete(cm, indexes, axis=1)
        cm_norm = np.delete(cm_norm, indexes, axis=0)
        cm_norm = np.delete(cm_norm, indexes, axis=1)

        classes = np.delete(classes, indexes, axis=0)

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

    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,
           ylabel='True label',
           xlabel='Predicted label',
           title="Precisión promedio = {0:.2f} %".format(np.mean(cm.
↪diagonal())) if clean else title)

    plt.setp(ax.get_xticklabels(), rotation=45, ha="right",␣
↪rotation_mode="anchor")

    fmt = 'd'
    fmt_norm = '.2f'

    thresh = 50

    if showLabels:
        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
```

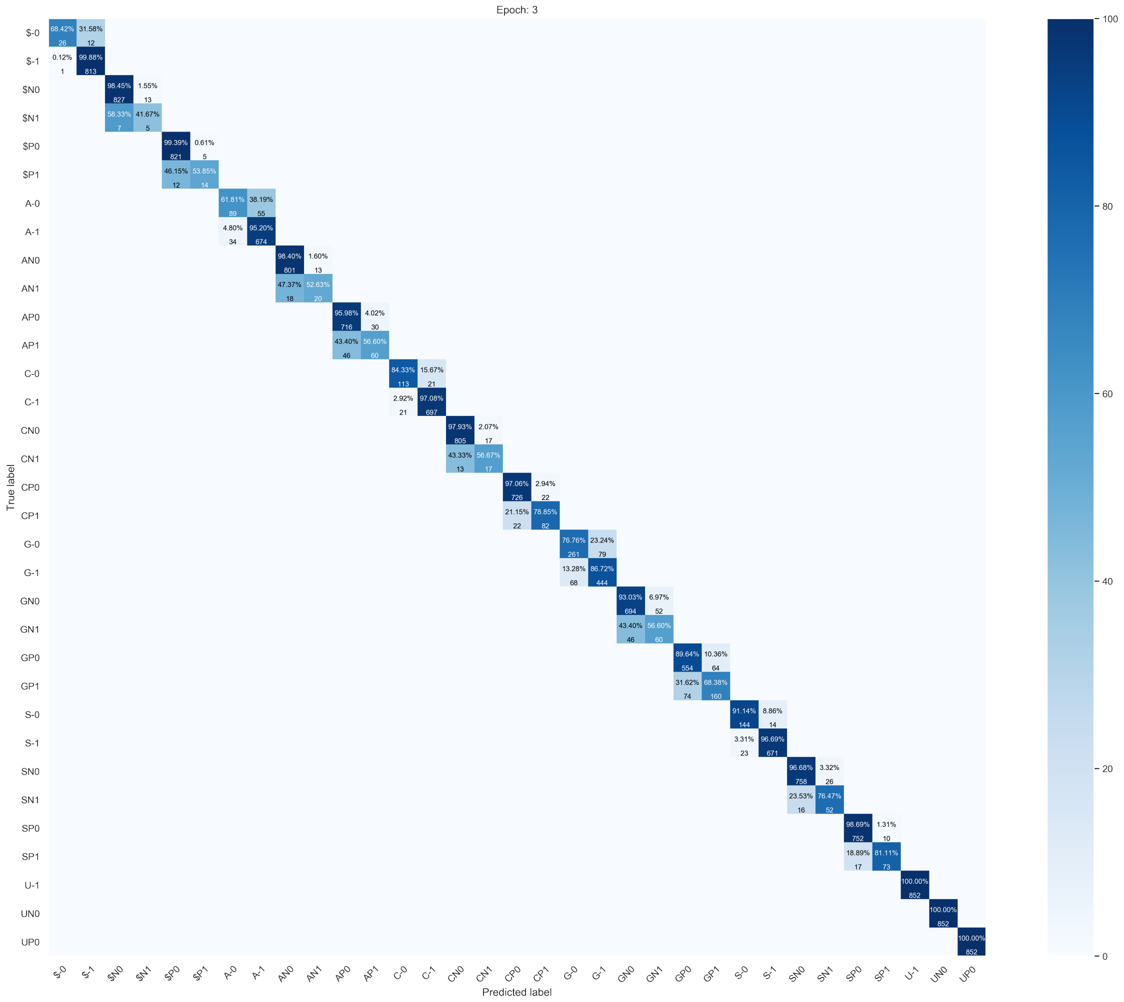
[8]:
```
y_real = test['y_real'].values
y_preds = {}

for k in test.keys():
    if 'y_label_pred_' in k:
        y_preds[k] = test[k].values
```

[11]:
```
for k in test.keys():
    if 'y_label_pred_' in k:
        y_pred = y_preds[k]
        k = k.replace('y_label_pred_', '')
        plot_confusion_matrix(y_real, y_pred, classes=unique_labels(y_real),␣
 ↪title="Epoch: {0}".format(k))
```

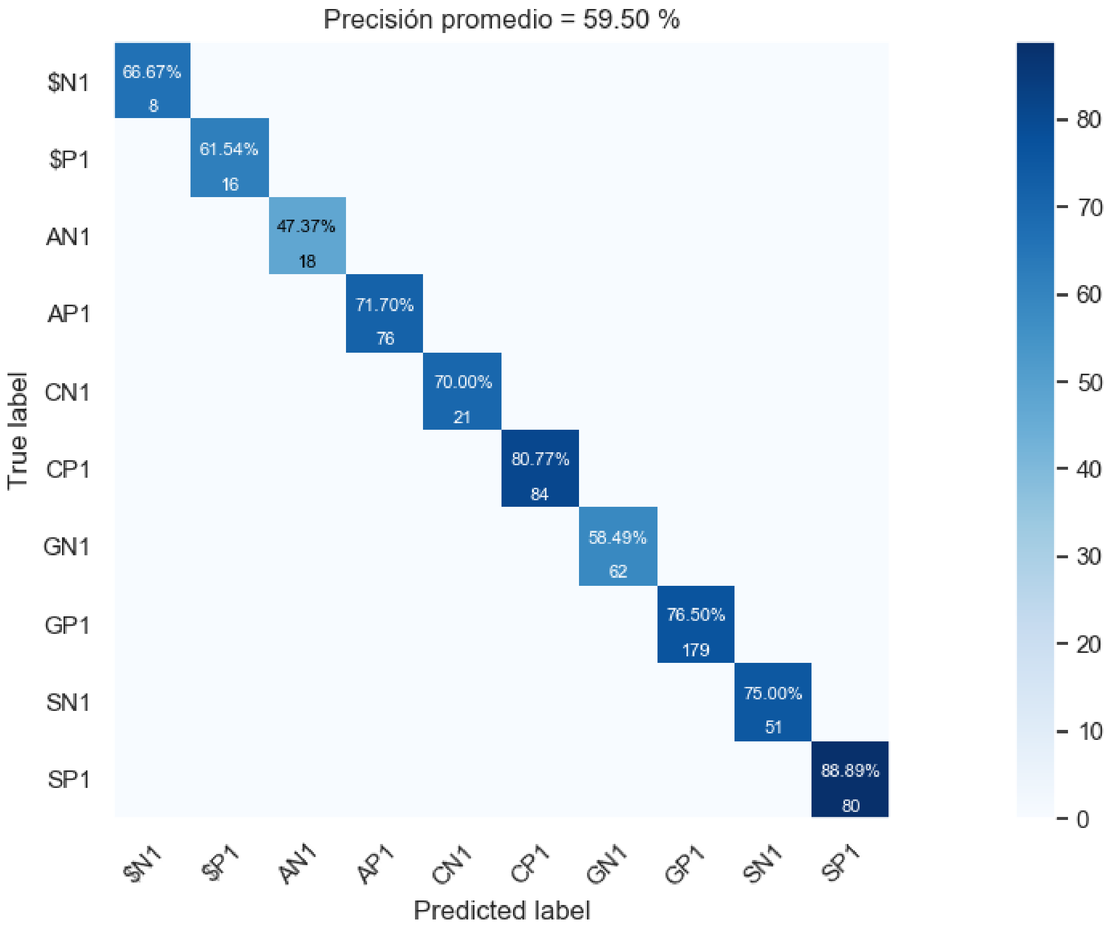


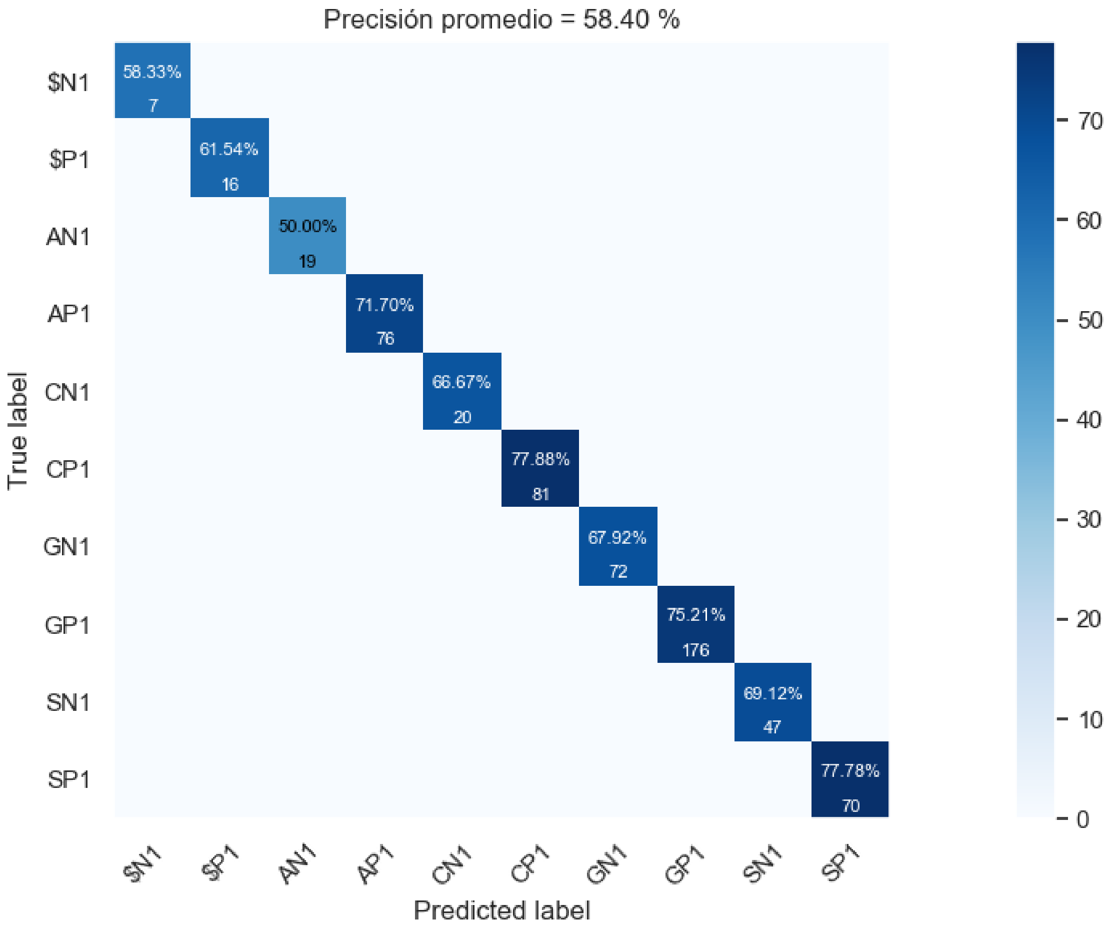

Epoch: 1
True label
$-0
65.79%
25
34.21%
13
$-1
100.00%
814
$N0
98.57%
828
1.43%
12
$N1
41.67%
5
58.33%
7
$P0
98.43%
813
1.57%
13
$P1
38.46%
10
61.54%
16
A-0
64.58%
93
35.42%
51
A-1
5.37%
38
94.63%
670
AN0
96.93%
789
3.07%
25
AN1
50.00%
19
50.00%
19
AP0
94.37%
704
5.63%
42
AP1
28.30%
30
71.70%
76
C-0
85.82%
115
14.18%
19
C-1
3.48%
25
96.52%
693
CN0
97.69%
803
2.31%
19
CN1
33.33%
10
66.67%
20
CP0
96.52%
722
3.48%
26
CP1
22.12%
23
77.88%
81
G-0
78.24%
266
21.76%
74
G-1
13.09%
67
86.91%
445
GN0
89.28%
666
10.72%
80
GN1
32.08%
34
67.92%
72
GP0
87.38%
540
12.62%
78
GP1
24.79%
58
75.21%
176
S-0
87.97%
139
12.03%
19
S-1
1.73%
12
98.27%
682
SN0
97.45%
764
2.55%
20
SN1
30.88%
21
69.12%
47
SP0
98.43%
750
1.57%
12
SP1
22.22%
20
77.78%
70
U-1
100.00%
852
UN0
100.00%
852
UP0
100.00%
852
$-0
$-1
$N0
$N1
$P0
$P1
A-0
A-1
AN0
AN1
AP0
AP1
C-0
C-1
CN0
CN1
CP0
CP1
G-0
G-1
GN0
GN1
GP0
GP1
S-0
S-1
SN0
SN1
SP0
SP1
U-1
UN0
UP0
Predicted label
100
80
60
40
20
0

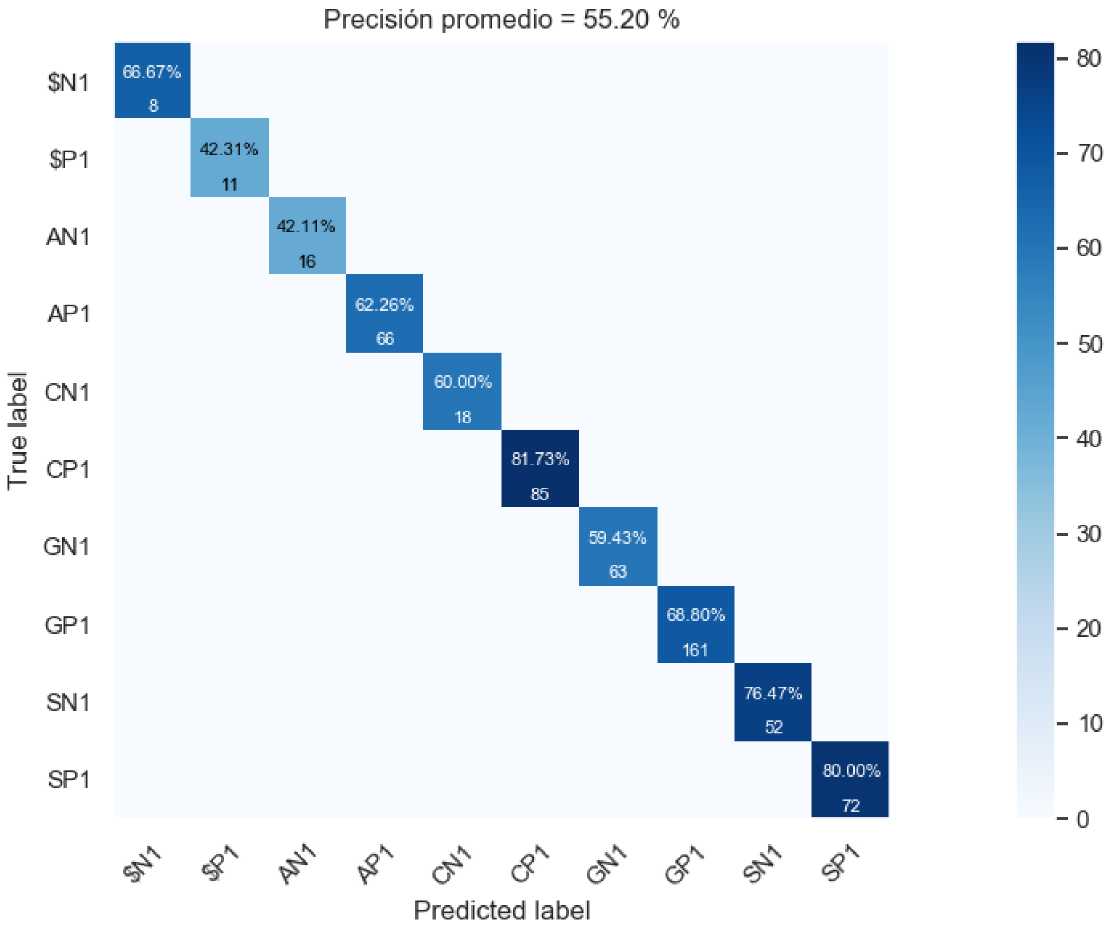

Epoch: 2
True label
$-0
$-1
$N0
$N1
$P0
$P1
A-0
A-1
AN0
AN1
AP0
AP1
C-0
C-1
CN0
CN1
CP0
CP1
G-0
G-1
GN0
GN1
GP0
GP1
S-0
S-1
SN0
SN1
SP0
SP1
U-1
UN0
UP0
63.16% 24
36.84% 14
100.00% 814
99.40% 835
0.60% 5
33.33% 4
66.67% 8
100.00% 826
57.69% 15
42.31% 11
61.11% 88
38.89% 56
5.08% 36
94.92% 672
98.40% 801
1.60% 13
57.89% 22
42.11% 16
94.10% 702
5.90% 44
37.74% 40
62.26% 66
90.30% 121
9.70% 13
3.62% 26
96.38% 692
97.93% 805
2.07% 17
40.00% 12
60.00% 18
96.26% 720
3.74% 28
18.27% 19
81.73% 85
77.65% 264
22.35% 76
9.77% 50
90.23% 462
93.43% 697
6.57% 49
40.57% 43
59.43% 63
89.00% 550
11.00% 68
31.20% 73
68.80% 161
90.51% 143
9.49% 15
2.59% 18
97.41% 676
98.21% 770
1.79% 14
23.53% 16
76.47% 52
97.64% 744
2.36% 18
20.00% 18
80.00% 72
100.00% 852
100.00% 852
100.00% 852
$-0
$-1
$N0
$N1
$P0
$P1
A-0
A-1
AN0
AN1
AP0
AP1
C-0
C-1
CN0
CN1
CP0
CP1
G-0
G-1
GN0
GN1
GP0
GP1
S-0
S-1
SN0
SN1
SP0
SP1
U-1
UN0
UP0
Predicted label
100
80
60
40
20
0

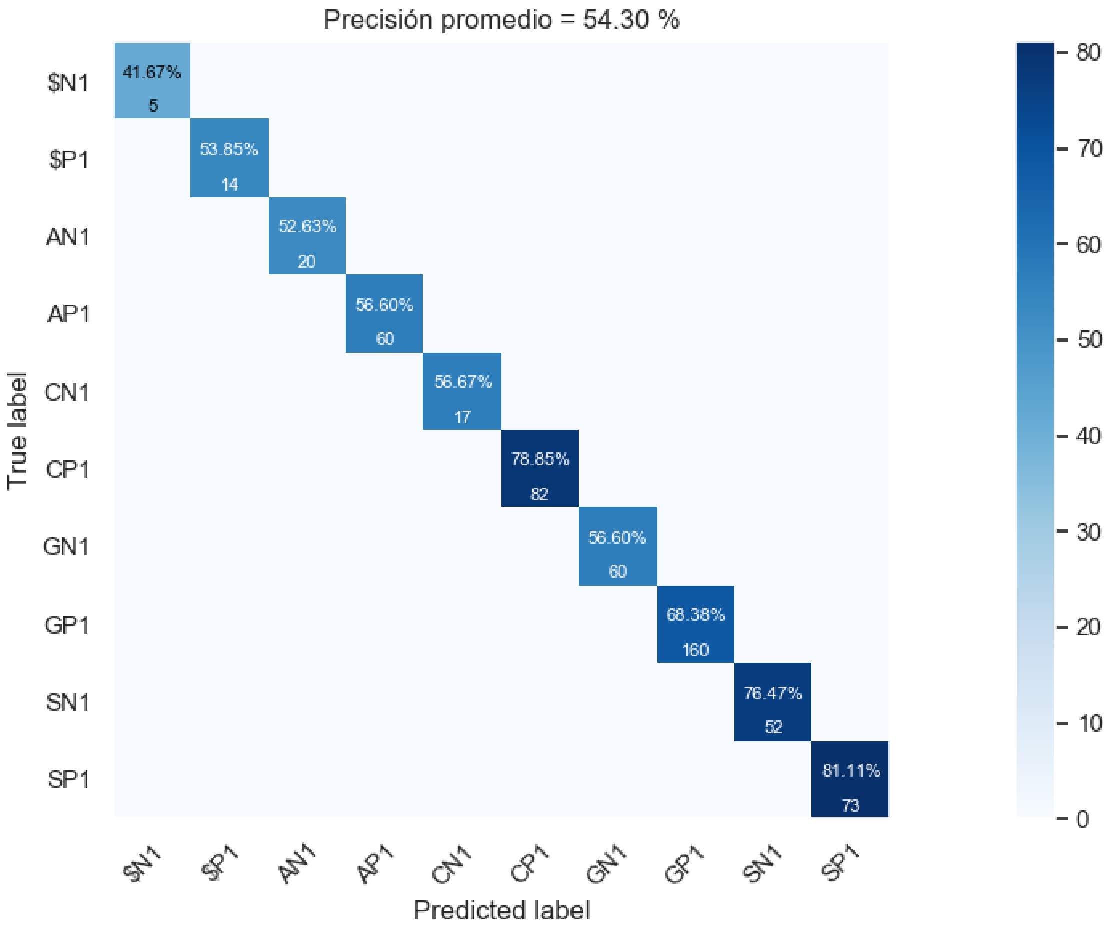


[10]:
```
for k in test.keys():
    if 'y_label_pred_' in k:
        y_pred = y_preds[k]
        k = k.replace('y_label_pred_', '')
        plot_confusion_matrix(y_real, y_pred, classes=unique_labels(y_real),␣
 ↪clean=True, figsize=(16, 6), dpi=100)
```


Precisión promedio = 59.50 %


Precisión promedio = 58.40 %
True label
$N1
$P1
AN1
AP1
CN1
CP1
GN1
GP1
SN1
SP1
58.33%
7
61.54%
16
50.00%
19
71.70%
76
66.67%
20
77.88%
81
67.92%
72
75.21%
176
69.12%
47
77.78%
70
$N1
$P1
AN1
AP1
CN1
CP1
GN1
GP1
SN1
SP1
Predicted label
70
60
50
40
30
20
10
0


Precisión promedio = 55.20 %
$N1
$P1
AN1
AP1
CN1
CP1
GN1
GP1
SN1
SP1
True label
66.67%
8
42.31%
11
42.11%
16
62.26%
66
60.00%
18
81.73%
85
59.43%
63
68.80%
161
76.47%
52
80.00%
72
$N1
$P1
AN1
AP1
CN1
CP1
GN1
GP1
SN1
SP1
Predicted label
80
70
60
50
40
30
20
10
0


Precisión promedio = 54.30 %
True label
Predicted label
$N1
$P1
AN1
AP1
CN1
CP1
GN1
GP1
SN1
SP1
41.67%
5
53.85%
14
52.63%
20
56.60%
60
56.67%
17
78.85%
82
56.60%
60
68.38%
160
76.47%
52
81.11%
73
80
70
60
50
40
30
20
10
0