Evaluacio?n para PFC

August 25, 2016

```
In [25]: import sys
         sys.path.insert(0, "../lib")
         import numpy as np
         import matplotlib.pyplot as plt
         %matplotlib inline
```

```
In [26]: def get_accuracy(y):
             y_real=y[:,0]
             y_pred=y[:,1]
             metrics= y_real==y_pred
             accuracy =(np.sum(metrics)*100.0)/len(metrics)
             #print "accruracy "+"{:10.4f}".format(accuracy) +"%"
             return "{:10.4f}".format(accuracy)+"%"
```

```
In [55]: from sklearn.metrics import precision_score
         from sklearn.metrics import recall_score
         from sklearn.metrics import confusion_matrix
         from sklearn.metrics import precision_recall_fscore_support
         from sklearn.metrics import f1_score
         def get_precision(y):
             y_real=y[:,0]
             y_pred=y[:,1]
             return " "+"{:10.4f}".format(precision_score(y_real,y_pred,average=’micro’))

         def get_recall(y):
             y_real=y[:,0]
             y_pred=y[:,1]
             return " "+"{:10.4f}".format(recall_score(y_real,y_pred,average=’micro’))

         def get_fscore(y):
             y_real=y[:,0]
             y_pred=y[:,1]
             return " "+"{:10.4f}".format(f1_score(y_real, y_pred, average=’micro’))

         def get_confusion_matrix(y):
             y_real=y[:,0]
             y_pred=y[:,1]
             return confusion_matrix(np.array(y_real),np.array(y_pred))
```

```
In [35]: import pickle
         path="/Users/luispeinado/Downloads/evaluate_seq2_vs_all.pkl"
         data_evaluate2= pickle.load( open( path, "rb" ) )
```

```
        def load_result(path):
            return pickle.load( open( path, "rb" ) )
```

```
In [18]: seq_model_label=['seq2_model']
         seq_dataset_label=['seq1','seq2','seq3','seq4','seq5']
```

```
In [60]: len_data_evaluate_model=len(data_evaluate2)



         for i in range(0,len_data_evaluate_model):
             len_dataset_evaluate=len(data_evaluate2[i])
             seq_for_model=data_evaluate2[i]
             print "model "+seq_model_label[i]
             result_line=""
             for j in range(0,len_dataset_evaluate):
                 result_line=" "+seq_dataset_label[j]+": "+(get_accuracy(np.array(seq_for_model[j])))+"
                 get_fscore(np.array(seq_for_model[j]))
                 print result_line
                 confusion_matrix= get_confusion_matrix(np.array(seq_for_model[j]))

                 plt.imshow(confusion_matrix, interpolation='nearest', cmap=plt.cm.Blues)
```

```
model seq1_model
```

```
        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-60-a8cc852bf732> in <module>()
         10     for j in range(0,len_dataset_evaluate):
         11         result_line=" "+seq_dataset_label[j]+": "+(get_accuracy(np.array(seq_for_model[j])))+
    ---> 12         +" recall "+(get_recall(np.array(seq_for_model[j])))
         13         get_fscore(np.array(seq_for_model[j]))
         14         print result_line


        TypeError: bad operand type for unary +: 'str'
```

```
In [103]: def evaluate(data_evaluate2,seq_model_label,seq_dataset_label,labels):
              len_data_evaluate_model=len(data_evaluate2)
              for i in range(0,len_data_evaluate_model):
                  len_dataset_evaluate=len(data_evaluate2[i])
                  seq_for_model=data_evaluate2[i]
                  print "model "+seq_model_label[i]
                  result_line=""
                  plt.figure(figsize=(12,12 ))
                  for j in range(0,len_dataset_evaluate):
                      result_line=" "+seq_dataset_label[j]+": "+(get_accuracy(np.array(seq_for_model[j]
                      print result_line
                      confusion_matrix= get_confusion_matrix(np.array(seq_for_model[j]))
                      plt.subplot(2,3,j+1)
                      plt.imshow(confusion_matrix, interpolation='nearest', cmap=plt.cm.Blues)
```

```
                    tick_marks = np.arange(len(labels))
                    plt.xticks(tick_marks, labels, rotation=90)
                    plt.yticks(tick_marks, labels)
                    plt.title(seq_dataset_label[j])
                    plt.tight_layout()
```

1 1 experimento todos contra todos

1.1 1.1 size of batch 25

Se testean todos las secuencias contra todas las secuencias con un batch de 25

```
In [83]: ### labels

         class_labels=['Corridor','Hall','ProfessorOffice','StudentOffice','TechnicalRoom','Toilet'
                       ,'SecretaryOffice','VideoConferenceRoom','Warehouse','ElevatorArea']
```

```
In [65]: path="../data/results/evaluate_all_vs_all.pkl"
         Y= load_result(path)
         type(Y)

Out[65]: numpy.ndarray
```

```
In [ ]:
```

```
In [104]: seq_model_label=['seq1_model','seq2_model','seq3_model','seq4_model','seq5_model']
          seq_dataset_label=['seq1','seq2','seq3','seq4','seq5']

          evaluate(Y,seq_model_label,seq_dataset_label,class_labels)
```

```
model seq1_model
 seq1:    27.2918% precision :      0.2729 recall      0.2729 f1_score      0.2729
 seq2:    26.7307% precision :      0.2673 recall      0.2673 f1_score      0.2673
 seq3:    27.2687% precision :      0.2727 recall      0.2727 f1_score      0.2727
 seq4:    26.4401% precision :      0.2644 recall      0.2644 f1_score      0.2644
 seq5:    17.6771% precision :      0.1768 recall      0.1768 f1_score      0.1768
model seq2_model
 seq1:    32.5241% precision :      0.3252 recall      0.3252 f1_score      0.3252
 seq2:    33.0421% precision :      0.3304 recall      0.3304 f1_score      0.3304
 seq3:    29.8043% precision :      0.2980 recall      0.2980 f1_score      0.2980
 seq4:    32.0763% precision :      0.3208 recall      0.3208 f1_score      0.3208
 seq5:    20.5540% precision :      0.2055 recall      0.2055 f1_score      0.2055
model seq3_model
 seq1:    30.0126% precision :      0.3001 recall      0.3001 f1_score      0.3001
 seq2:    29.2422% precision :      0.2924 recall      0.2924 f1_score      0.2924
 seq3:    30.0267% precision :      0.3003 recall      0.3003 f1_score      0.3003
 seq4:    28.5744% precision :      0.2857 recall      0.2857 f1_score      0.2857
 seq5:    19.4365% precision :      0.1944 recall      0.1944 f1_score      0.1944
model seq4_model
 seq1:    31.0590% precision :      0.3106 recall      0.3106 f1_score      0.3106
 seq2:    32.1468% precision :      0.3215 recall      0.3215 f1_score      0.3215
 seq3:    30.5605% precision :      0.3056 recall      0.3056 f1_score      0.3056
 seq4:    31.3925% precision :      0.3139 recall      0.3139 f1_score      0.3139
 seq5:    20.1498% precision :      0.2015 recall      0.2015 f1_score      0.2015
```

```
model seq5_model
 seq1:    20.5525% precision :      0.2055 recall      0.2055 f1_score      0.2055
 seq2:    20.8779% precision :      0.2088 recall      0.2088 f1_score      0.2088
 seq3:    19.6619% precision :      0.1966 recall      0.1966 f1_score      0.1966
 seq4:    21.4463% precision :      0.2145 recall      0.2145 f1_score      0.2145
 seq5:    15.4422% precision :      0.1544 recall      0.1544 f1_score      0.1544
```

Evaluacio\begingroup \let \relax \relax \endgroup [Pleaseinsert\P

Evaluacio\begingroup \let \relax \relax \endgroup [Pleaseinsert\P

Evaluacio\begingroup \let \relax \relax \endgroup [Pleaseinsert\P

Evaluacio\begingroup \let \relax \relax \endgroup [Pleaseinsert\P

Evaluacio\begingroup \let \relax \relax \endgroup [Pleaseinsert\P

```
In [ ]:
```