

Bimodal IEMOCAP Documentation

Contents

Module bimodal_iemocap	1
Functions	1
Function Bimodal	1
Function calc_test_result	1
Function get_report	1
Function get_scores	2
Function plot_clf_report	2
Function plot_confusion_matrix	2
Function plot_tag_scores	3

Module bimodal_iemocap

Functions

Function Bimodal

```
def Bimodal()
```

load unimodal activations saved at “../input/multimodal-sentiment/unimodal.pickle” The pickle file must contain a dictionary of numpy arrays having the feature vectors with keys: ‘audio_train’, ‘video_train’, ‘audio_test’, ‘text_train’, ‘test_mask’, ‘test_label’, ‘video_test’, ‘train_mask’, ‘text_test’, ‘train_label’ Trains bimodal layers using the input feature vector(text and audio only). The bimodal model layers are defined herein Prints statistics of prediction on test set

Returns: final: a 3D numpy array of predictions of model in one hot encoded form (test_size by sequence_length by number of classes) test_label: a 3D numpy array of true labels in one hot encoded form (test_size by sequence_length by number of classes) test_mask: a 2D numpy array telling which inputs to ignore while calculating accuracies

Function calc_test_result

```
def calc_test_result(
    result,
    test_label,
    test_mask
)
```

Generates various classification stats in terminal

Args: result: a 3D numpy array of predictions of model in one hot encoded form (test_size by sequence_length by number of classes) test_label: a 3D numpy array of true labels in one hot encoded form (test_size by sequence_length by number of classes) test_mask: a 2D numpy array telling which inputs to ignore while calculating accuracies

Returns: None

Function get_report

```
def get_report(
    y_true,
```

```

        y_pred,
        classes
    )

```

This function parses the classification report given by sklearn to get all the row names metric values as floats and supports for each class label.

Args ---= **y_true** : true (numerical) labels of data

y_pred predicted (numerical) labels of the same data

classes a python list of class labels

Returns ---= **class_names** : a python list of class labels (here, row names from report)

plotMat numerical values (metrics) in the classification report

support the number of instances for each class_name present in report

Function get_scores

```

def get_scores(
    y_true,
    y_pred,
    classes
)

```

This function calculates the correct and incorrect counts for each label as a fraction to the total instances of that class.

Args ---= **y_true** : true (numerical) labels of data

y_pred predicted (numerical) labels of the same data

classes a python list of class labels

Returns ---= numpy array of tuple of (correct,incorrect) fractions for each class

Function plot_clf_report

```

def plot_clf_report(
    classes,
    plotMat,
    support,
    cmap=<matplotlib.colors.LinearSegmentedColormap object>
)

```

This function plots the classification report as an image, using the parsed values from the sklearn classification report and saves the image in the current working directory.

Args ---= **classes** : a python list of class labels

plotMat numerical values (metrics) in the classification report

support the number of instances for each class present in report

cmap the color map to be used in the output image

filename the filename with which the plot will be saved (can be a path too)

Returns ---= No return value. Shows and saves the report image.

Function plot_confusion_matrix

```

def plot_confusion_matrix(
    classes,
    mat,
    normalize=True,
    cmap=<matplotlib.colors.LinearSegmentedColormap object>
)

```

This function plots the confusion matrix as an image, using the parsed values from the confusion matrix and saves the image in the current working directory.

Args ---= **classes** : a python list of class labels

mat numerical values (metrics) in the confusion matrix

normalize controls the normalization of the confusion matrix (rows sum to 1 or not)

cmap the color map to be used in the output image

filename the filename with which the plot will be saved (can be a path too)

Returns ---= No return value. Shows and saves the confusion matrix image.

Function `plot_tag_scores`

```
def plot_tag_scores(  
    classes,  
    scores,  
    normalize=True  
)
```

This function plots the histogram for tag scores and saves the image in the current working directory.

Args ---= **classes** : a python list of class labels

scores a dictionary of correct and incorrect counts for each label

filename the filename with which the plot will be saved (can be a path too)

Returns ---= No return value. Shows and saves the tag scores plot.

Generated by *pdoc* 0.9.2 (<https://pdoc3.github.io>).