Unimodal MOSEI Documentation

Contents

Module `unimodal_mosei`

Functions

Function `Unimodal_text`

```
def Unimodal_text()
```

Loads unimodal activations saved at “../input/mosei-dataset-3way/text_3way.pickle” The pickle file must contain a 2 element list of numpy arrays with the first element having the feature vectors and the second with the labels Trains unimodal layers using the input feature vector. The unimodal 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) model: The final trained model

Function `calc_test_result`

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

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)

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 `loadGloveModel`

```
def loadGloveModel(
    File
)
```

Loads word embeddings from specified file Args: File: file path of word embeddings

Returns: gloveModel: dictionary containing key, value pairs of word and its embedding as an numpy array

Function `main_func`

```
def main_func()
```

Contains variable stmt, change it to see the output of any example sentence. Runs and trains the model, prints and saves the confusion matrix, tags scores, reports Args: None

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).