Homework 5:
Sentiment Analysis and Perceptron

Dr. Benjamin Roth
Computerlinguistische Anwendungen

Due: Thursday November 29, 2018, 16:00

In this exercise you will implement a Perceptron Classifier that will be trained on a data set of movie reviews to classify them into *positive* and *negative* reviews.
Take a look at the file `hw05_perceptron/perceptron.py`. In this exercise you will have to complete some methods to make the classification work.

This homework will be graded using unit tests (see details on last exercise sheet, including additional hidden tests). You can run the unit tests with:

```
python3 -m unittest -v hw05_perceptron/test_perceptron.py
```

Note: Make sure to have a folder called *data* under your source folder. We also put some useful data structures that you know from previous exercises into `hw05_perceptron/utils` you can use if you want to.

Exercise 1: Constructing the Classifier [4 Points]

Exercise 1.1: From a File

Complete the classmethod `PerceptronClassifier.from_file(cls, filepath)` that initializes a perceptron classifier parting from a (*json*-) file with a given filepath.

Exercise 1.2: From a Dataset

Complete the classmethod `PerceptronClassifier.from_dataset(cls, dataset)` that creates a perceptron classifier with initial (untrained, neutral) weights for the features in a given data set.

Exercise 2: Predicting [4 Points]

Complete the method `PerceptronClassifier.prediction(self, counts)`. This method should return a either 1 or -1 indicating the classification. (POS: `+1`, NEG: `-1`)

Exercise 3: Perceptron Update [4 Points]

During training the weights of the perceptron classifier are iteratively being adjusted by performing the so called *perceptron update*:

- If the classifier's prediction for a training instance is already correct, do nothing.
- Otherwise increase/reduce the weights of your classifier so that they better fit the training data. Keep in mind that negative weights are possible.

Complete the method `PerceptronClassifier.update(self, instance)`.
In this method you only have to replace two lines:

- Replace `error = 0` with the correct calculation of the error.
- In the for loop replace the `pass` statement with the correct update of feature weights.

Exercise 4: Using the classifier [4 points]

Once you have implemented all missing functionality, complete `sentiment.py` in order to train the classifier on a dataset of actual movie reviews.

- We will use a set of movie reviews from nltk as our corpus. In order to use the corpus, you have to first download it. For this, execute the following lines in the interactive python3 interpreter:
  ```
  >>> import nltk
  >>> nltk.download()
  ```
  In the downloader, download either the `book` collection or the `movie_review` corpus.
- In `sentiment.py`, have a look at `load_reviews` and understand how the data is loaded and split into training and test.
- Now, change the code to enable training in `nltk_movie_review_accuracy`:
  1. Remove the `return` statement in the first line (this skipped loading the data so that the tests would fail faster initially).
  2. Replace the `pass` statement with the correct call to the training method.