Homework 5:
Sentiment Analysis and the Perceptron

Dr. Benjamin Roth
Computerlinguistische Anwendungen

Due: Thursday December 7, 2017, 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. **Note:** We put some useful data structures that you know from previous exercises into `hw05_perceptron/utils`.

This homework will be graded using unit tests (see details on last exercise sheet). You can run the tests with:
`python3 -m unittest -v hw05_perceptron/test_perceptron.py`

Exercise 1: Constructing the Classifier for a Dataset [4 Points]

Complete the classmethod `PerceptronClassifier.for_dataset(cls, dataset)` that creates a perceptron classifier with initial (untrained) 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 boolean value indicating the classification. (POS: `True`, NEG: `False`)

Exercise 3: Perceptron Update [4 Points]

During training the weights of the perceptron classifier are iteratively 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 use a corpus that is already part of nltk. 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.