

Programming Assignment 5 (due **Thursday 21 April 2022**)

Write a python program called `sentiment.py` that adapts your `wsd.py` program to perform (positive/negative) sentiment analysis over [tweets](#) [Download tweets](#) from twitter.

Your program should **use the bag-of-words features from your wsd plus three additional features** of your choosing. Please make sure you only identify features from the training data, and that you clearly explain what features you are using in your detailed comments because you will be adding on to these in subsequent assignments.

Your classifier should run from the command line as follows:

`python3 sentiment.py sentiment-train.txt sentiment-test.txt my-model.txt > my-sentiment-answers.txt`

This command should learn a model from `sentiment-train.txt` and apply that to each of the sentences found in `sentiment-test.txt` in order to assign a sentiment to each of the tweets. Do not use `sentiment-test.txt` in any other way (and only identify features from `sentiment-train.txt`). Your program should output the model it learns to `my-model.txt`. You may format your model as you wish, but please make sure to show each feature, the log-likelihood or probability scores associated with it, and the sentiment it predicts. The file `my-model.txt` is intended to be used as a log file in debugging your program. Your program should output the sentiment it creates for each sentence to STDOUT. Your sentiment should be in the same format as found in `sentiment-key.txt`.

`sentiment-train.txt` contains examples of the tweets where the correct sentiment is marked in the text (to serve as an example from which to learn). `sentiment-test.txt` contains tweets without any sentiment being indicated, where the correct answer is found in the file `sentiment-test-key.txt`. You can find `sentiment-train.txt`, `sentiment-test.txt` and `sentiment-test-key.txt` in the assignment posting on Canvas.

You should also write a utility program called `scorer.pl` which will take as input your sentiment tagged output and compare it with the gold standard "key" data (`sentiment-test-key.txt`). Your scorer program should report the overall accuracy of your tagging, and provide a confusion matrix. This program should write output to STDOUT.

The scorer program should be run as follows:

`python3 scorer.py my-sentiment-answers.txt sentiment-test-key.txt`

You can certainly use your `scorer.py` program from the previous assignment as a foundation for this program.

Both `sentiment.py` and `scorer.py` should be documented according to the standards of the programming assignment rubric.

In `sentiment.py`: include what model you implemented, your accuracy and confusion matrix into the comments. And compare your results to that of the most frequent sense baseline.

Please submit your program source code for both `sentiment.py` and `scorer.py` to the Canvas. Please remember to upload the files individually *not* as a zip file.