HW1 Code Submission Guidelines
COMS W4701 Spring 2018

Please read this guideline fully and carefully
to ensure your submission can be graded correctly by our system.

The programming portion of HW1 will be graded automatically using a script. On CourseWorks, you should submit one file, `hw1_<your_uni>.zip`.

When unzipped with the command `unzip hw1_<your_uni>.zip`, the resulting directory must contain the following files:

1. `4_1.py`, which produces a new file: `ner_train_rare.dat`, the data file with `_RARE_` words. We will then run `count_freqs.py` on the `ner_train_rare.dat` to produce the counts file `ner_rare.counts`.
2. `4_2.py`, which produces `4_2.txt`, the tagged `ner_dev.dat` data with the extra log likelihood column.
3. `5_1.py`, which produces `5_1.txt`, the file containing trigrams and their respective parameters in the format, "w_{i-2} w_{i-1} w_i $log_e q(w_i|w_{i-2}, w_{i-1})$". Assume that the file to be read will be called `trigrams.txt`, and will be located in your root directory. Each line of this `trigrams.txt` contains the trigrams: "w_{i-2} w_{i-1} w_i".
4. `5_2.py`, which produces `5_2.txt`, the tagged `ner_dev.dat` data in the same format as `4_2.txt` but tagged using the Viterbi tagger.
5. `6.py`, which produces `6.txt`, the tagged `ner_dev.dat` data in the same format as `4_2.txt` and `5_2.txt` but tagged using the your improved Viterbi tagger that better deals with rare words.
6. `<your_uni>_observations.pdf`, a pdf containing your answers to the written portions of the coding problems. Your name and uni should be at the top, followed by the sections "Question 4", "Question 5", and "Question 6", each containing your observations/discussions for the respective problems.

The directory should also contain the original hw files: `count_freqs.py`, `eval_ne_tagger.py`, `ner_train.dat`, `ner_dev.dat`, and `ner_dev.key`, as well as any additional files your scripts need to run correctly. If you do use additional files, they may organized in subdirectories as you like, but make sure to use hardcoded relative paths to these files.

The first line of each script should begin with a shebang `#!/usr/bin/python` or `#!/usr/bin/python3`. This will allow the files to be run from command line using `./<some_script>.py`. We recommend that you test your scripts by running them from command line using this syntax before submitting to be sure we can run them and they produce the required outputs as intended. Note that you may need to add execution permissions with `chmod +x <some_script>.py`.

If you don't write your code in Python (for example in Java, Perl, CPP, etc), you still have to provide a simple python wrapper with the above names. For example, if you have java code `4_1.java`, then your submitted `4_1.py` script may contain the following lines:

```
#!/usr/bin/python
import os,sys
os.chdir(os.path.dirname(sys.argv[0]))    # Going to the current directory.
```

```
os.system('javac 4_1.java')
os.system('java 4_1')
```

Finally, **DO NOT** use any external libraries unless specified in the assignment. The grading system won't have them installed and your code will not run. Allowed Packages: `numpy`, `scipy`, `sklearn`, `pandas`.