CIS 4930 NLP // HW #3 // Spring 2018

Date Assigned: January 25, 2018
Date Due: January 31, 2018

Submission Format

You will submit a soft copy of your solution using e-Learning (http://elearning.ufl.edu) by the end of the day (23:59 / 11:59 PM) on the assigned date (January 31). Save and submit two files, **hw3.pdf** and **hw3.py**. Include the *analysis* part of your solution in the **pdf** file and the python *code* you implement in the **py** file.

Assignment

At the top of every solution file you submit this semester include: your name, section number, the assignment number, and the date due. Complete these exercises. In your answers, you may find it useful to write some code, run a test, and report the result of the test. In addition to reporting test results, analyze your results and assert why you have drawn the conclusions given in your answers.

Exercises

- **2.4**: Read in the text of the *State of the Union* address, using the `state_union` corpus reader. Count occurrences of `men`, `women`, and `people` in each document. What has happened to the usage of these words over time?

- **2.13**: What percentage of noun synsets have no hypnoyms? You can get all noun synsets using `wn.all_synsets('n')`.

- **2.19**: Write a program to create a table of word frequencies by genre, like the one given in (http://www.nltk.org/book/ch02.html#sec-extracting-text-from-corpora) for modals. Choose your own words and try to find words whose presence (or absence) is typical of a genre. Discuss your findings.

- If you have played around with the example code in the textbook, you may have discovered that Moby Dick is an extremely lexically diverse book. Some of this diversity may not be appropriate for all audiences. Implement a function *censor* that receives a list of bad words, a list of more favorable words, and a string of text, and returns the string with all of the occurrences of the bad words replaced with the corresponding words from the better list. I would like to see some iteration in your solution, so avoid the string 'replace' method and regular expressions; you are welcome to use 'split' and 'join' - you can assume that

strings given to this method have no punctuation or have spaces separating all words and punctuation. Here is a sample function call:

```
>>> bad_words = [ 'hell', 'Queequeg', 'dumpling' ]
>>> better_words = [ 'heck', 'Tony', 'core' ]
>>> sample_text = 'In one word , Queequeg , said I , rather
digressively ; hell is an idea first born on an undigested apple -
dumpling'

print( censor( bad_words, better_words, sample_text ) )
```