CSCE 355, Spring 2019, Homework 1
Due 4 February 2019

The point of this assignment is to give you the opportunity to write much of the basic code for dealing with text as it will exist in this class.

Your assignment is to read multiple files from two different genres of writing, do a frequency count of reporting verbs, and then write a brief commentary about whether or why the counts in the two genres are the same (I don't think they will be) or are different. For the final analysis, you should probably look at the top 10 (or maybe 20) most frequent reporting verbs.

Frequency in this context is defined as the frequency per 1000 words, so this normalizes raw counts by the size of the underlying text base.

You have a list of "reporting verbs" in the zip file for the assignment. These are the verbs used in making claims, assertions, arguments, etc. This list is probably best read into a Python `set` for later efficient lookup. You also have a list of the CLAWS part-of-speech tags. You won't have to read these in, but these will be useful to look at now and to use in the future.

You should use as your text the documents in the "2014_4_101_400" data set provided of student data, as one genre, and the first 20 documents in the "00" set of the 2009 COCA academic data, as the other genre. (I think this is roughly comparable in size; if not, use as much COCA data as is necessary to get roughly the same word counts. I will before class on Wednesday the 23d do this program and be able to update the needs, but I wanted to get this text out to you now and not delay.)

Use the `sentpos` files of CLAWS-tagged sentences.

You will need to do several things to get this working. None of these is hard; each is a standard function used in statistical text processing.

- You will need to read multiple files from different directories. For this, the Python `listdir` function is useful, because it gives you a list of the file names in a given directory, and you can loop over the file names in the returned list.

- You should read the `sentpos` files, throw away the first items in each line that are not text and part of speech, and read the part of the lines that is text and POS.

- You should do a frequency count of total number of tokens and of the various reporting verbs. These are each probably best done with

a Python `defaultdict(int)` that has the word as the key and the frequency as the value.

- Some of the reporting verbs (like "report", curiously enough) could be either a noun or a verb. You will need to read the "B" lines of the `sentpos` files, which will give you `word_POS` tokens, split those tokens on the underscore, and then check that the word part of the token is in the reporting verbs set and that the POS part of the token starts with "V". (Note that in the CLAWS 7 tag set, verbs and only verbs have a POS tag that starts with a "V". For this purpose, you can hard-code the insistence on starting with "V". For more sophisticated things, you'd have to do a more complicated data structure or list of things to be looked for.)

Finally, I would like perhaps one page (max of three) of an analysis of the results. I have not actually done this study for this data, but on other data in the larger corpora I have found that the lists of most frequent reporting verbs is different for student writing than it is for more formal academic writing (which is what this particular COCA data is).

If you normalize per thousand tokens, then you need to think about what you are counting as a token. A more naive version of this might go ahead and count punctuation tokens as tokens. If the two genres had roughly the same number of words, and the same number of sentences (and thus the same number of words per sentence), then one could perhaps argue that they had the same number of punctuation tokens per sentence. A more precise counting would exclude the punctuation tokens from the overall counts. This would be the reverse of the starts-with-V that identifies words that are reporting verbs and are tagged as verbs and not nouns; you'd have a set of punctuation tags and not count if the POS was in that set.

I'm not going to insist that you do all the fancy stuff to get to a totally clean result, but I would expect you to discuss these caveats in your analysis of what the results might indicate. The programming part of this includes several steps each of which should be relatively straightforward for students at your level. Perhaps the more important part is recognizing that it's ok in science not to have conclusive results and answers as long as you are clear in your writing that you know you don't (yet) have all the results and answers.