Matt Luettgen
Assignment 2
Note: all the code is on a Github repository, https://github.com/mrluettg/LING-L415 (Let me know if you can't access this).

1. For this assignment I will compare English written by a native speaker and English written by a native Japanese speaker. I am choosing *The Strange Case of Dr. Jekyll and Mr. Hyde* by Robert Louis Stevsenson to represent the native English and *A Japanese Boy* by Shiukichi Shigemi to represent the non-native English. These all come from Gutenberg. I chose these books in particular because they were both published around the late 1890s and both contain around 30k words.

a. and b. - These were done with the file *n_gram_frequency.py*. This file works in the command line by typing "py n_gram_frequency.txt [file_name] [number of n grams]" and will write the file [filename]_1_frequency.txt (You cannot write a file using '>,' with this because of unicode mapping reasons). For part a, see "jekyll_hyde_frequency_1.txt" and "japanese_boy_shiukichi_frequency_1.txt." For part b, see "jekyll_hyde_frequency_2.txt" and "japanese_boy_shiukichi_frequency_2.txt"

c. This was calculated using *type_token_ratio.py*, which works by typing "py type_token_ratio.py [frequency_file]" in the command line and prints the type-token ratio. The type token ratio for *Jekyll and Hyde* is 0.1314 and the type token ratio for *A Japanese Boy* is 0.1689. This means that Shiukichi's lexicon was slightly more diverse than Stevenson's, though they are both similar.

d.

I may be interpreting the equation wrong (many math symbols, some I don't know what it means), but I *think* it's telling me to choose a document, find add up all the tags in that document, and then divide it by the number of tags in the entire corpus. I've added some more documents from Gutenberg to make a more sizeable corpus: *Call of the Wild* by Jack London, *Around the World in Eighty Days* by Jules Verne, and *The Secret Agent* by Jospeh Conrad. I calculated generality using the file *calc_generosity.py*. This works by typing the first file being compared and the rest of the corpus in the command line, all frequency files created by *n_gram_frequency.py*: "py calc_generality.py [compare_file] [corpus_files]+" so for this case, the specific line I typed was "py calc_generality.py japanese_boy_shiukichi_1_frequency.txt call_wild_london_1_frequency.txt jekyll_hyde_stevenson_1_frequency.txt secret_agent_conrad_1_frequency.txt eighty_days_verne_1_frequency.txt"

It produced the following output:

```
Tags in document: 6141
Tags in corpus: 17600
```

Generality: 0.34892045454545456

e.

This was calculated with the python script *top_ten_bigrams_no_stopwords.py*. This works by entering one of the 2_frequency files (the bigram ones) created by *n_gram_frequency.py* as “py top_ten_no_stopwords.py [frequency_2 file]” I also added a list to make sure that punctuation was not included. I also added “‘s” to this blacklist because I think it’s pretty much a function word, though the nltk stopwords list did not include it.

Here is the result for A Japanese boy.

```
new year 19
mr. gladness 11
young man 8
miss chrysanthemum 8
one end 7
aunt otsuné 6
takes place 5
india ink 5
main street 5
next morning 4
```

This list makes a lot of sense. The top result is “new year” which has more cultural significance in Japan than it does in the U.S. There are a lot of character names such as “mr. gladness,” “miss Chrysanthemum,”and “aunt otsuné,” as well as place names such as “main street,” and “india ink.” There are also a few common phrases such as “one end,” “takes place,” and “main street.”

The results for jekyll and hyde are:

```
mr. utterson 71
mr. hyde 32
henry jekyll 29
dr. jekyll 26
edward hyde 25
said mr. 18
said poole 11
mr. enfield 11
dr. lanyon 10
said utterson 9
```

Interestingly, the stopwords in nltk can evidently detect the difference between the period in an abbreviation and the period at the end of a sentence. Again, here we see many character names, and in significantly larger amounts (the jekyll and hyde document has about 25k words and the Japanese boy has about 30k). A possible reason why this may happen is that Shigemi’s native language is Japanese, which tends to be more contextual than English, often dropping subjects and objects if they can be understood from the context. Another interesting feature is that Stevenson’s list contains a lot of the word “said.” This likely indicates a difference in the two authors’ writing styles.

f. the bigrams I am selecting from the japanese boy doc are “mr. gladness” (these two are do not outside of each other and appear many times, so they should be high for all the tests), “of the” (two very common words - should be very low), and indiscreet juvenile (these two only appeared next to each other once and each only appeared once by themselves, so they should be high). The file I used to do this was *stats.py*, which works as “py stats.py [title_author] word_1 word_2”, so in the case of the bigram “of the” in the *A Japanese Boy* document, it would look like “py stats.py japanese_boy_shiukichi of the.” The second mutual information is the real mutual information. The first is calculated using Yang and Peterson’s method.

Results for “of the”

```
log-likelyhood: 9.587578723835598
mutual_information: 1.3657245261357611
mutual_information_2: 1.2366060040931
X2:  764.6105577319494
```

Results for C “mr. gladness”

```
log-likelyhood: 32.413757121773514
mutual_information: 8.103439280443379
mutual_information_2: 12.649695203062057
X2:  36364.0
```

Results for “indiscreet juvenile”

```
log-likelyhood: 42.005338212967
mutual_information: 10.50133455324175
mutual_information_2: 17.445485748658797
X2:  36364.00000000001
```

The final two have the highest, though I think the fact that these words only appear with each other causes some odd numbers, mainly that their X2 scores are nearly equal and that “indiscreet juvenile” has higher mutual information and log-likelihood than “mr. gladness,” despite appearing less frequently. They are definitely both better than “of the” in all tests, which makes sense.