CMPSC 102
Discrete Structures
Fall 2018

Practical 11:
Who Says It Like That?!
Word Frequencies to The Rescue

Refer to your notes, slides and sample Python code from this week and other weeks. In particular, follow the python code that we created in class or check on line for interesting pieces of code to help you in your programming.

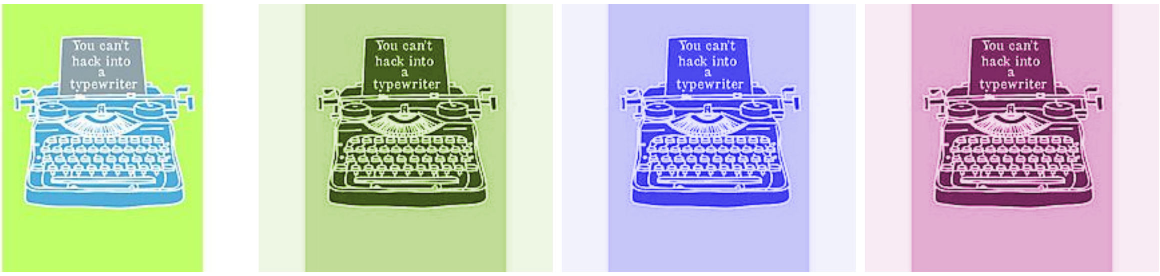
You can't hack into a typewriter
You can't hack into a typewriter
You can't hack into a typewriter
You can't hack into a typewriter

Figure 1: Maybe we cannot hack into a typewriter but we can still hack the text that typewriters have produced. For this type of hacking, we collect frequency information to determine the distribution of frequencies.

GitHub Starter Link

`https://classroom.github.com/a/r0L9Yk-E`

To use this link, please follow the steps below.

- Click on the link and accept the assignment.
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab.
- Clone this repository (bearing your name) and work on the practical locally.
- As you are working on your practical, you are to commit and push regularly. You can use the following commands to add a single file, you must be in the directory where the file is located (or add the path to the file in the command):

  – `git commit <nameOfFile> -m ‘‘Your notes about commit here’’`

  – `git push`

  Alternatively, you can use the following commands to add multiple files from your repository:

- `git add -A`
- `git commit -m ''Your notes about commit here''`
- `git push`

Summary

You are to run an analysis of word frequencies across several different authors to determine a form of resemblance between them. Then you will modify the given code to output several additional types of frequency plots to get potentially better views for a comparison of results of the text.

I Wonder Why They Are So Similar?!

Have you even wondered why the music of some rock-bands always seems to have the same kinds of *sounds* in their music? Or, have you ever wondered why film directors always have the same kinds of themes in their films? Or, have you spoken to a person about two different types of subjects, only to realize that they described their ideas of the subjects *in-sort-of-the-same-way*? Finally, have you ever read two or more books by the same author, and found that the books were very similar to each other (for some reason), even though the plots were actually quite different? *Dude, what gives?*

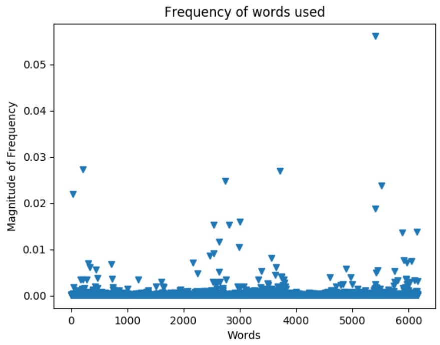

Figure 2: The frequency plot of Author Conan Doyle's *Hound of the Baskervilles.*

It could be that these similarities are present because of human bias in communication. *Cool*, but where do these biases come from? A person speaks (or writes) with a language style, a particular syntactic structure and a selected lexicon of words with which to describe ideas. Here, the notion that one's language can be used to determine one's authorship has been recently explored the literature by Hoover [**?**] who explored word usage (called *N-gram* distributions) and Hu *et al.* [**?**], who investigated one's selected lexicons of language. The sentiment of language has even been studied by Culpeper *et al.* to determine the influence of William Shakespeare's macabre writing styles that may have drifted from his tragedies to those written by John Webster [**?**].

These differences of language usage may be studied and are characterized by the natural biases of language. In this practical, we will investigate a simple technique to study bias in written communications.

Today's Work

In Figure 1, the idea that you cannot hack a typewriter is raised, however the production of its texts is widely open to your investigation. In today's practical, you will use provided code to investigate the frequencies of word usage in efforts to determine an author's linguistic *fingerprint* for the comparison to the fingerprints of other authors. For this task, you are to use the included Python code (File: `src/wordSearcher_11.py`) to calculate the frequencies of words in public-domain novels to determine whether the frequencies of the same author are similar.

Run Your Code and Perform Your Analysis

Experiment with the given literary works: Included with your code are four text files of public domain novels. There are two Sherlock Holmes works (*of course!*) by Author Conan Doyle: (*The Hound of the Baskervilles* and *The Speckled Band*) and, also two works by Jane Austen (*Pride and Prejudice* and *Sense and Sensibility.*) You are feed these files into your code to obtain frequency plots of each work for similarity analysis. Your plots will resemble the plot of Figure 2.

1. **Run your code**: Use the below to help you run the program located in `src/wordSearcher.py` of your starter repository.

   ```
   python3 wordSearcher.py textfile.txt
   ```

   The program will show and save a plot, like that of Figure 2 for each work.

2. **Get other works for analysis**: Next, you are to report to the Project Gutenberg website `http://www.gutenberg.org/` to download other free books of your choosing as text files. Be sure to get two or more literary works from several different authors for your analysis of frequency plots. For the same author's works, what kinds of similarity do you see in his or her works? What kinds of dissimilarity do you find in the author's works? Now, find another author and check the all the frequency plots again for similarity and dissimilarity?

3. **Add two new plotters**: Report to the Matplotlib examples website at `matplotlib.org/gallery/index.html` to get ideas and associated code for at least two additional plotting functions for your code to use.

4. **Questions in blue**: Place your Markdown-written responses to these questions in the file: `writing/miniReflection.md`.

1. **The Source Code**: In your own words, how does the program work to find frequencies from text files for plotting?

2. **The Selected Data**: What authors and literary works did you choose?

3. **Comparisons in Same Author**: What similarities and dissimilarities between the same author did you find?

4. **Comparisons of Different Authors**: When compared to another author, what kinds of noticeable similarities and dissimilarities did you uncover?

Deliverables

1. Your completed reflection document (`writing/miniReflection.md`) where you respond to the questions in blue (above).
2. Your complete (and working) code in `src/wordSearcher_11.py` that has been altered to include at least two other types of plots of the data.

```
.- -. .- --. .-. .- -- ...
```