Hunter Schilb and Teresa Wolf

Problem Solving and Software Design

Professor Li

30 October 2016

# Assignment #3: Text Mining and Analysis

For this assignment, we decided to conduct an analysis on different horror novel writers. We wanted to compare people's opinions of the scariest authors against the results of our text analysis. The goal of our analysis was to determine the scariest author by scanning their work for keywords from our “Scary Word List”. The author who had the highest number of scary words in their work would be the winner. The scary word list, which we downloaded as a text file, was chosen at random from a website. The authors we chose to research included Edgar Allan Poe, Henry James, HP Lovecraft and Ambrose Bierce. For Poe we chose *volumes 1,3 and 5 of the Work of Edgar Allan Poe,* for Henry James we chose *The Turn of the Screw*, for HP Lovecraft *The Shunned House*, and Ambrose Bierce’s *Occurrence at Owl Creek* and *the Damned Thing*.

The data structure we chose to house the scary word list in was a list (Variable name = scaryWords). The data structures we chose to house the contents of the books were histograms. The data structure we used to hold our results of our comparison was a dictionary. The program begins by returning a list of scary words from the scary word list. From there we created a function where the files of various authors could be uploaded, stripped of stopwords/punctions/whitespace, and other irrelevant text. Whatever remained after was appended to a list. We made sure to skip the gutenberg header and to stop running the code when it reached the footer of the document in order to analyze only the text written by the authors.

Our program then creates a histogram that returns a list of the most common words in the texts and their frequencies. The following function prints that list to then be analyzed by the “compare” function - keeping only the scary words we have defined in our scary word list. The program sums the frequencies of the scary words and the total is their “scary score”. Our code underneath the “#Analysis” comment runs the functions for each author’s text files. For authors with more than one text file, we took an average of their scary scores. Finally, the computer judges who the scariest author is by calculating which author has the largest scary score.

From a reflection standpoint, it wasn’t too difficult to set-up the necessary functions for use in our analysis portion. What we struggled with was being able to differentiate between the various data structures we could use. This included making decisions between histograms, nested lists, and dictionaries. Secondly, we had a difficult time translating strings to integers. To solve this we utilized office hours, researched online and spent hours experimenting different ideas with the code. We learned how to properly display our findings in human readable format. Specifically, we had the computer know what the contents of the large histograms are for each work, but in our final function getScariestAuthor(), we do print the histograms and only display the score for each author shown as an integer. We did this so that our final output is clean and simple, which means that it is easy to read without extra scrolling.

## Text Files Used

List of Authors:

*Edgar Allan Poe*

http://www.gutenberg.org/ebooks/2147

http://www.gutenberg.org/ebooks/2151

http://www.gutenberg.org/ebooks/2149

*HP Lovecraft*

http://www.gutenberg.org/ebooks/31469

*Henry James*

http://www.gutenberg.org/ebooks/209

*Ambrose Bierce*

http://www.gutenberg.org/ebooks/375

http://www.gutenberg.org/ebooks/23172

Resources:

Halloween Key Words: http://www.enchantedlearning.com/wordlist/halloween.shtml

***We pledge that we have neither received nor provided any unauthorized assistance during the completion of this work***