Eric Cao, Kevin Garcia

CAP 5571

12/11/2019

Final Project Report

**The Classification of Books**

**Abstract:**

For our final project we decided to do a classification problem, specifically classifying books acquired from Project Gutenberg[[1]](http://www.gutenberg.org/). By using word dictionaries found from SCOWL[[2]](http://wordlist.aspell.net/), we did association analysis to cluster books from the database based on age appropriateness, racial hate, and neither of the two. Note that the two categories, age appropriateness and racial hate, are not mutually exclusive, as some adult books could contain racial slurs and vice versa. Additionally, we looked at which word commonly precedes or follows another word. We also made a training set and testing set on the books to predict which category the next book an author writes would belong to. We also created graphs and tables to help visualize the number of books each category has.

**Introduction:**

Everyday more books are released, more news reported on, and more literature to read. With so many books to read, it is hard to figure out the book’s category without reading it. We decided to create a classification program in Python, and it will classify books based on the following categories: Adult, Hate, and Normal from SCOWL[[2]](http://wordlist.aspell.net/) dictionaries.

The program will classify books into the adult section if it contains inappropriate words, similar to the MPAA[[3]](https://www.motionpictures.org/film-ratings/) rating for movies; if a book contains a swear word, then the book will be classified into this category. The program will classify books into the hate section if it contains racially demeaning terms or slurs in the book; if the book has a racial slur, then the book will be classified into this section. The program will classify books into the normal section if it does not contain any swear words or racial slurs. Another dictionary we used was for function words[[4]](https://semanticsimilarity.wordpress.com/function-word-lists/). Function words are words that have little lexical meaning or have ambiguous meaning and express grammatical relationships among other words within a sentence or specify the attitude or mood of the speaker. Some examples of function words are “the”, “a”, “otherwise”, and “whether.” The four dictionaries can be found in the “Dictionary” folder.

We also wrote a program that looks at words, specifically which word precedes or follows another word. We wanted to see if there were any words that was commonly found next to each other. We then made tables and graphs to help visualize the data we found. We wrote the program in Python because it has libraries that makes classification and visualization easier. We found online word dictionaries from SCOWL[[2]](http://wordlist.aspell.net/) to classify books into the three categories. We found the books on Project Gutenberg[[1]](http://www.gutenberg.org/), an online database that contains over 60,000 books. For this project, we will be classifying 100 books. Some related works we found were not specifically about books. One paper at Cornell University where they detected hate speech in social media[[5]](https://arxiv.org/abs/1712.06427).

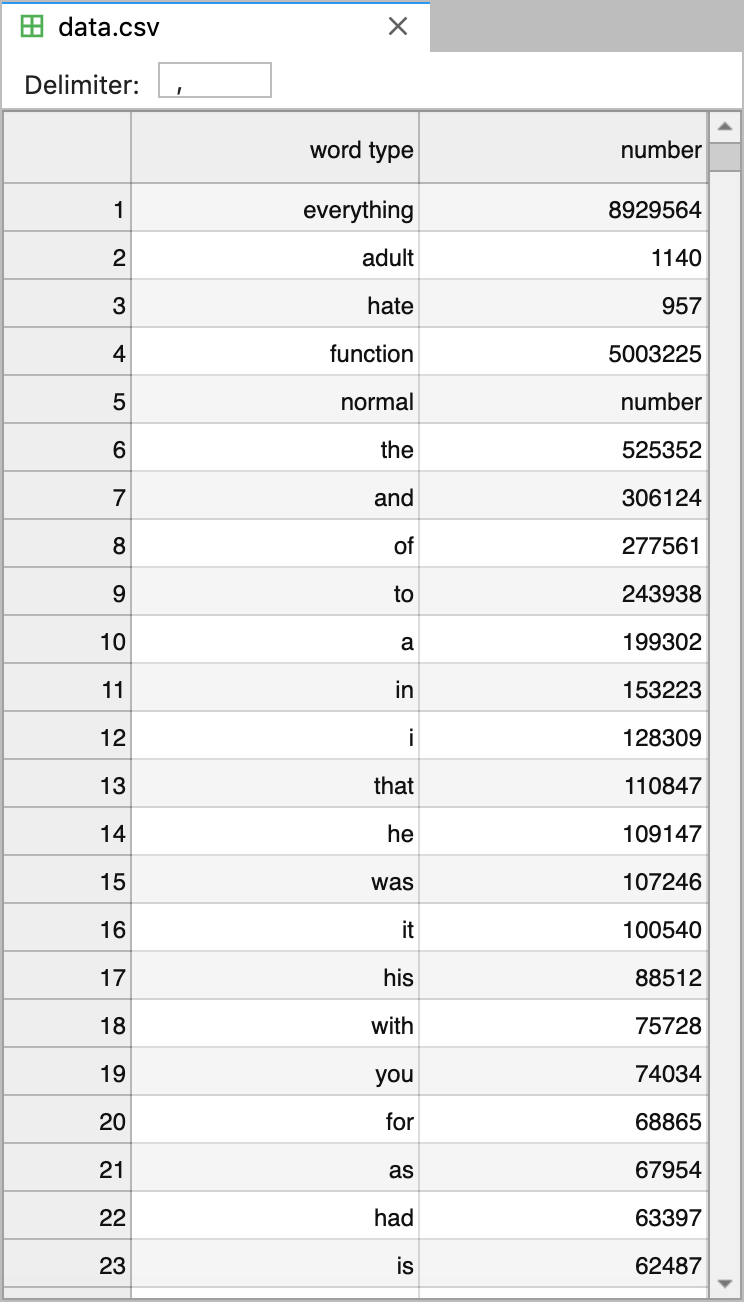
**Experiments/Methodology:**

For this experiment, we first downloaded the books from Project Gutenberg[[1]](http://www.gutenberg.org/) as text files to set up our dataset. Using various libraries, we parsed the books based on words, then counted how many times the word appears in the book, and then outputted that information into a csv file. Before putting into a csv file, we had to clean the data. That meant we had to remove extraneous characters, such as dashes and underscores, as well as numbers and made up words. We also made all the words the same case. The file has two columns: the first column contains the word and the second column contains the amount of times that word appeared in the book. We applied this procedure for all the books, for each author, and for each book. For before and after, the dataset looks different. The first column is the target word, the second column is the most common word before or after the target word, followed by its percentage in the next column. After that, we repeat this process twice for the second most common and the third most common.

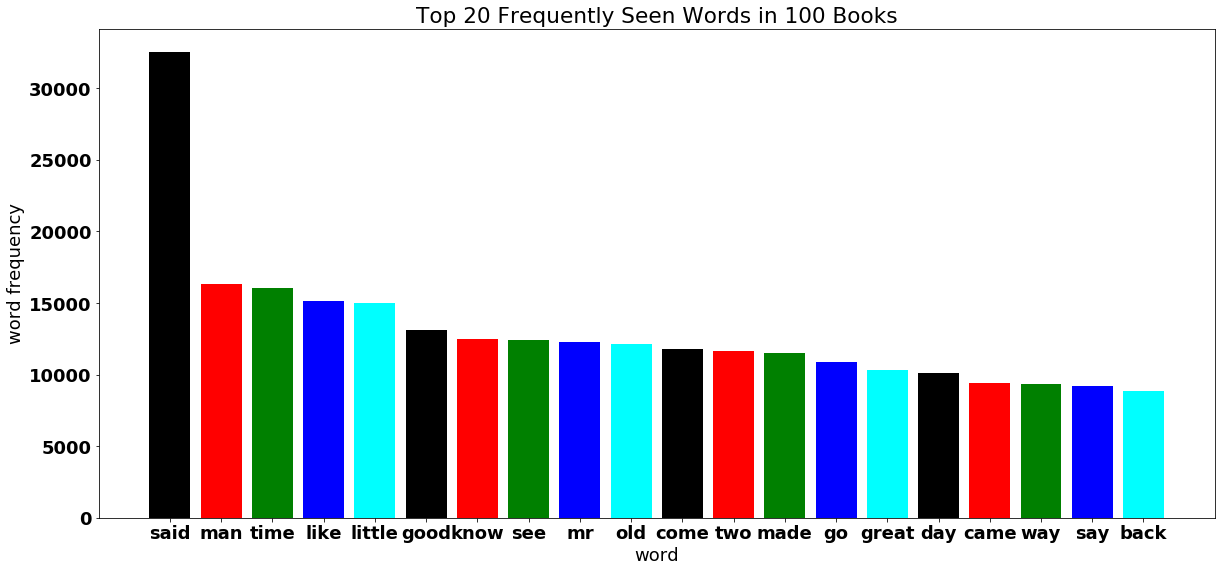
To make sure our program classified the books correctly, we split the dataset into a training set and test set and performed training and testing. In order to increase accuracy, we randomized the split of the dataset each epoch. We evaluated the program as follows: if the program thinks it found a book with hateful language and it does, then hate correct percentage goes up, and if it was wrong then the percentage goes down. This process iterates 1000 times before we get the final correct percentage. The program was also able to reasonably classify future books. For example, the program might think that a book written by a specific author has an 85% chance to contain adult language. We then clustered the books based on their classification and created tables and graphs to visualize the data. The programs were written in Python and the graphs and charts were written using the matplotlib library.

**Results:**

After running it, the program correctly detected hate speech 71.23% of time and correctly detected adult language 62.05% of the time, and this program checked 100 books and did 1000 iterations. Note that the values will change per run. Here are screenshots of some of the bar charts of the data.



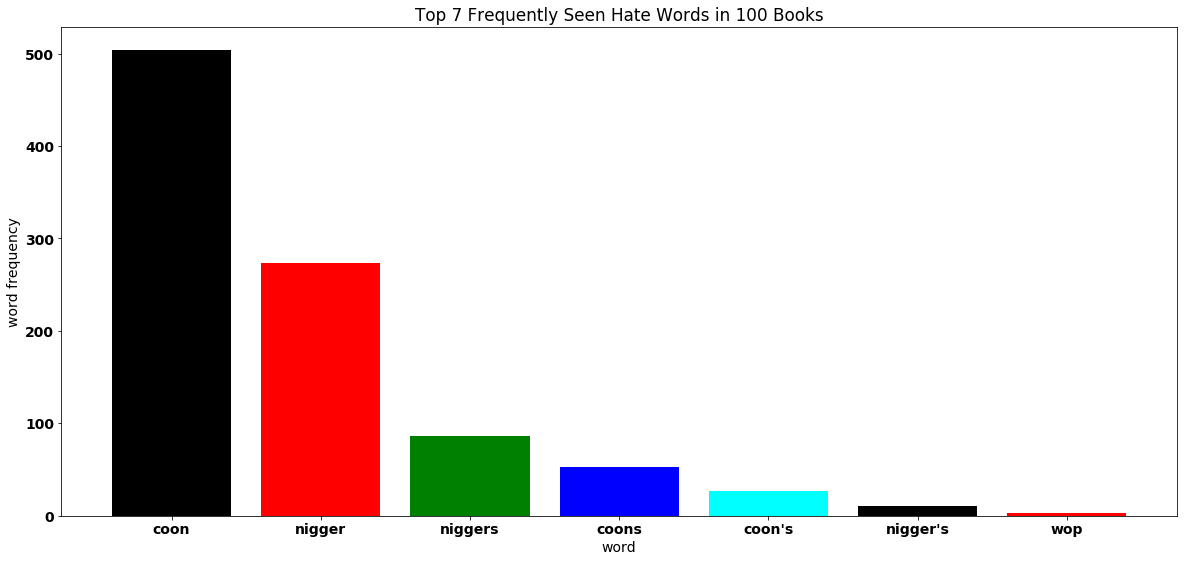
*Screenshot of the total word count of all books. The file is called data.csv*



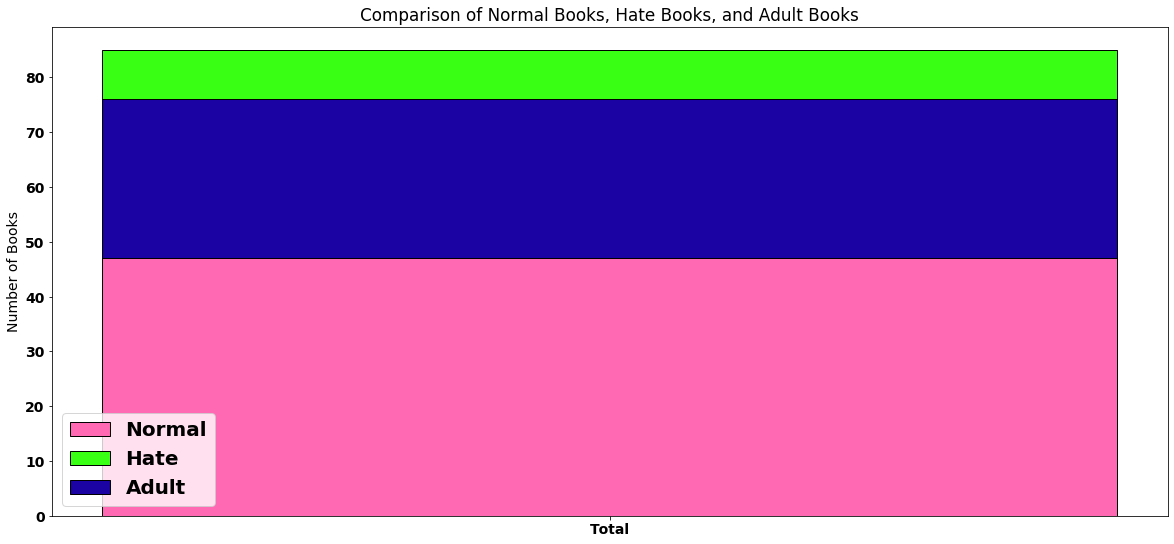
*Screenshot of bar-chart of the 20 most frequent words excluding function words*

**

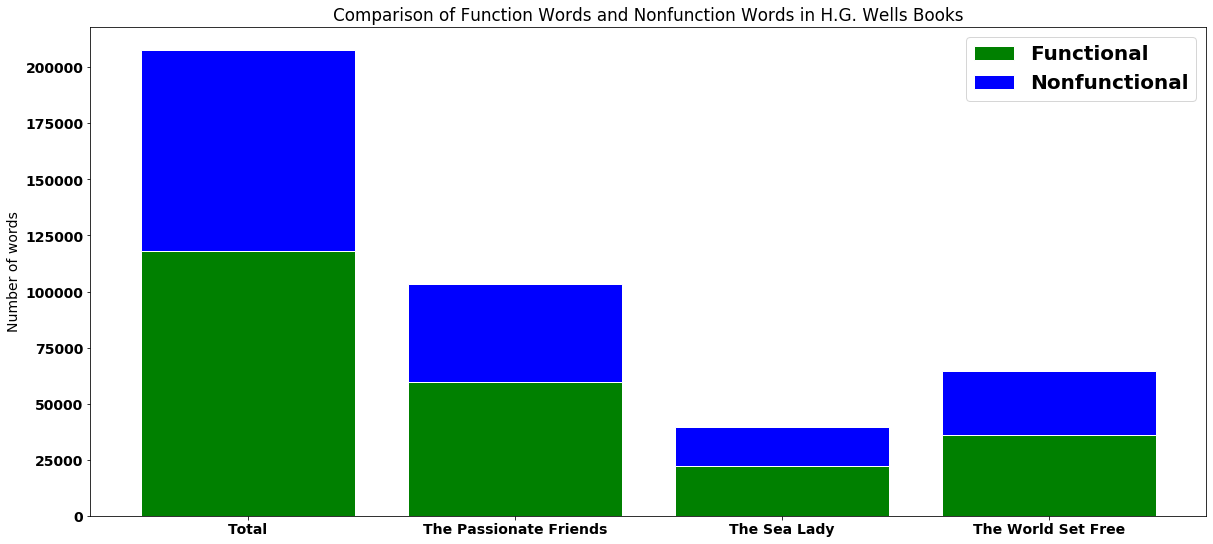
*Screenshot of bar-chart of the 20 most frequent adult words*

**

*Screenshot of bar-chart of the 7 most frequent hate words*

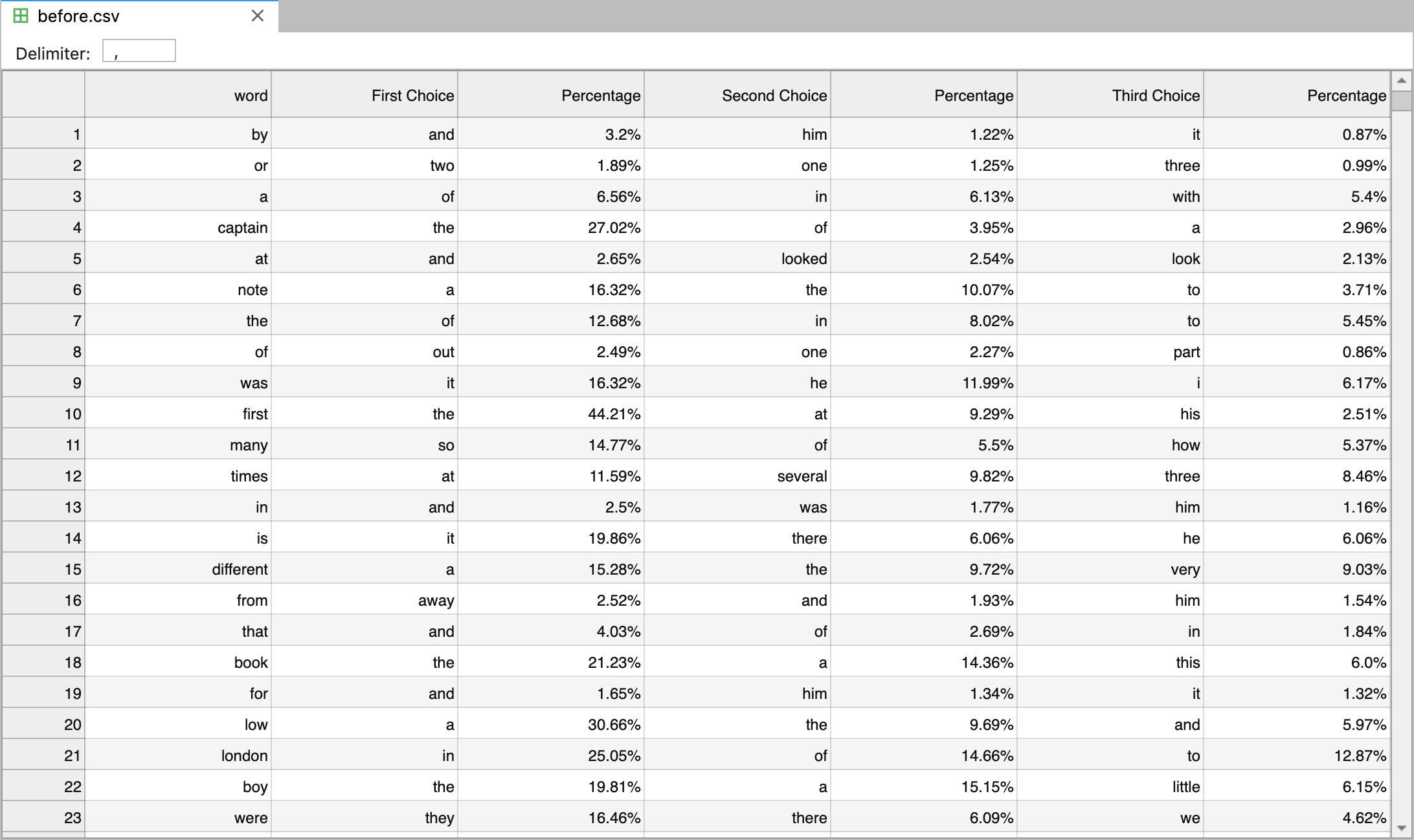
**

*Screenshot of stacked bar-chart showing the different book types*

**

*Screenshot of stacked bar-chart for the three books and the author displaying functional vs nonfunctional*

After reviewing the graphs, we noticed that the word “said” was the most common nonfunctional word. For the adult language dataset, “dick” was the most common word, and for the hate speech dataset “coon” was the most common word. Whereas “the” was the most common word overall. Overall, there were 3,924,242 non-functional words, 5,003,225 functional words, 957 hate words, 1140 adult words, resulting in a total of 8,929,564 words in 100 books.



*Screenshot of the dataset before.csv*

A screenshot of a cell phone

Description automatically generated

*Screenshot of histogram showing percentages that target word was preceded by its most common word*

*A picture containing screenshot

Description automatically generatedScreenshot of histogram showing percentages that target word was preceded by its most common word. Target word had to appear 1000 times at least.*

The histograms above show the correlation between the target word and the most frequently seen preceding word. As you can see, most of the words in the top histogram have high association analysis, and this is proven with having the most common word preceding the target word 90% to 100% of the time. After removing the words that have a word count less than 1000, we can see that there is a low association analysis with words, with the word “prince” preceding the word “andrew” 78.27% of the time. We did this process for both books and authors, and we did this for words following the target word.

**Conclusion:**

The reason why both “coon” and “dick” were the most common was because there were books that heavily skewed the data. One such book was Moby Dick. Another observation we noticed was that when we were making the datasets, some of the datasets included parts of the book that wasn’t present in other books, like prefaces, table of contents, or acknowledgements. This would skew some of the datasets as it had more different words. Another observation we had was for the histograms. The data at first shows high association analysis between words. But after we remove words that do not have a word count higher than 1000, we can see that there really is a low association analysis. Another issue we noticed was that some of the same authors had different names in Project Gutenberg. This resulted with misinformation, because our data would show that two different authors may have used adult language, but it was actually just one author who used adult language. Also, that they didn’t get grouped together, thus our data for each individual author was incomplete.

For future work, we can improve by fixing the issues mentioned above. Additionally, we can use this classification data to help us improve language modeling for certain authors, which will help people in the machine learning field. Another thing to improve would be the dictionaries, and our way of classifying adult books and hate books. The reason for this is because some insulting words and homonyms or homographs. An example of this is the word “dick.”

**References:**

**[1]**<http://www.gutenberg.org/>

**[2]**<http://wordlist.aspell.net/>

**[3]**<https://www.motionpictures.org/film-ratings/>

**[4]**<https://semanticsimilarity.wordpress.com/function-word-lists/>

**[5]**<https://arxiv.org/abs/1712.06427>