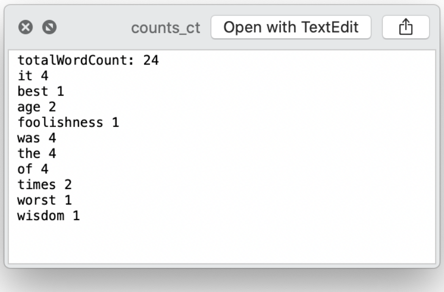
Project 6: Word Frequency

Abstract

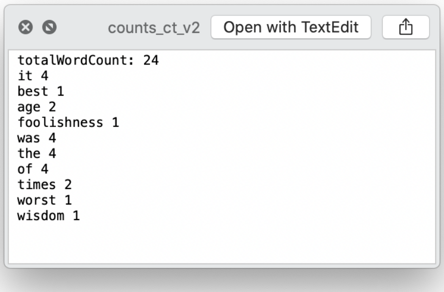
For this week's project, I used the BSTMap to determine the word frequencies of all words in a give text document. The BSTMap is an implementation of a map using a binary search tree (BST). A Map structure is a data structure that maps the key to a value, and a Binary Search Tree is a binary tree that keeps the keys in a sorted order. The BSTMap implements the MapSet interface, and I also created a KeyValuePair to hold the data on each node in the BSTMap. I also created a word count file for each text file of reddit comments from 2009-2015 to analyze when I ran the WordCounter on them. The word count file provided me with information of total word counts and the frequencies of each unique word appeared in the text file. At last, I plotted a scatter diagram to see the trend over the past 8 years.

Results

By running `diff counts_ct.txt counts_ct_v2.txt`, I saw no output, which means the two files are identical.

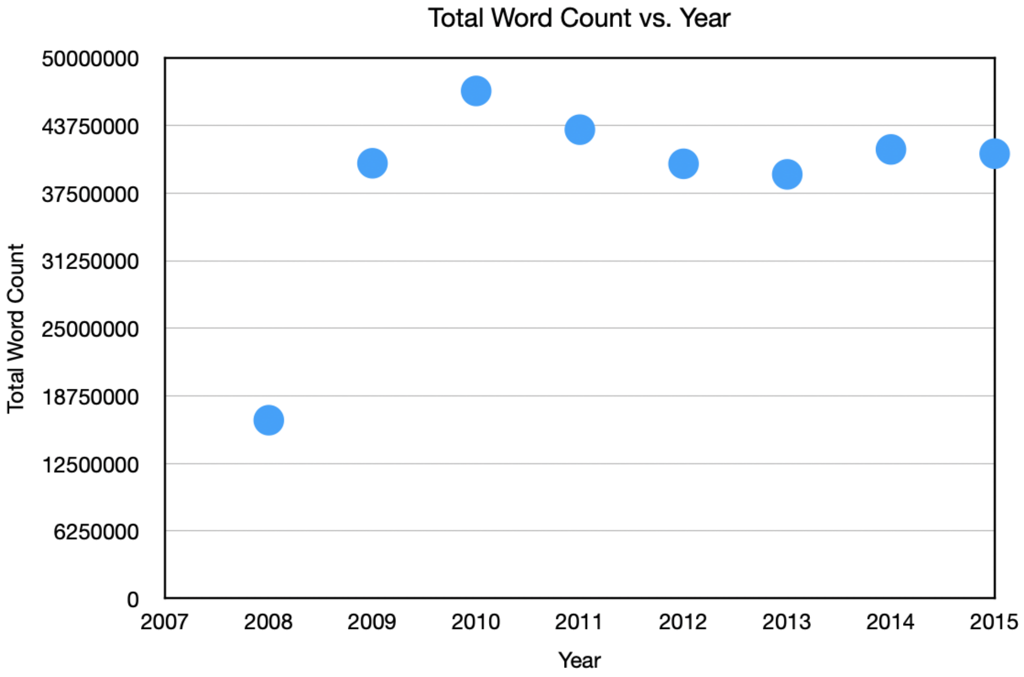


```
totalWordCount: 24
it 4
best 1
age 2
foolishness 1
was 4
the 4
of 4
times 2
worst 1
wisdom 1
```



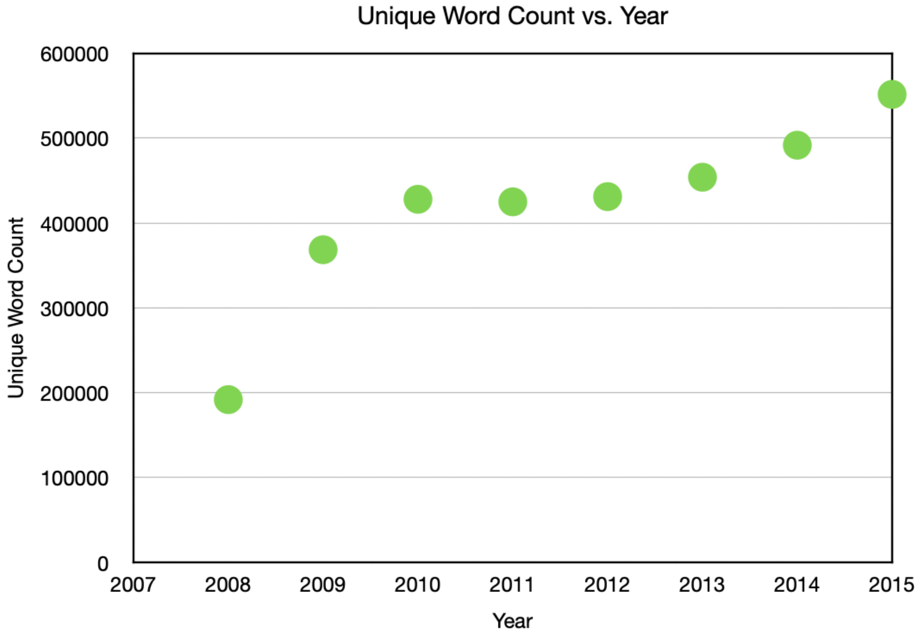
```
totalWordCount: 24
it 4
best 1
age 2
foolishness 1
was 4
the 4
of 4
times 2
worst 1
wisdom 1
```

The following graph shows the relationship between total word count and year of the text file.



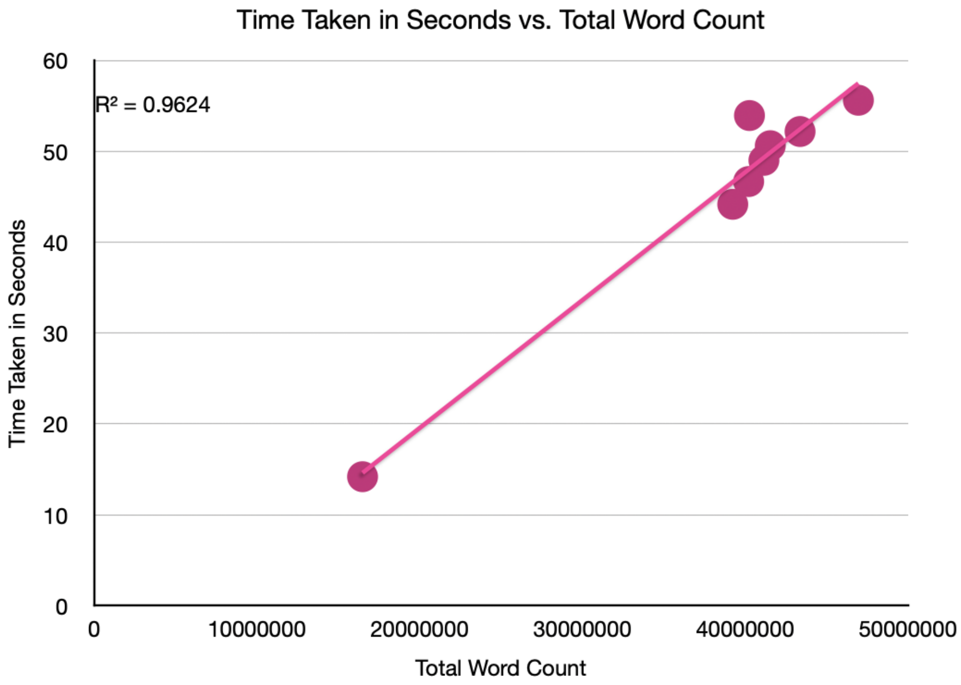
We can see a speedy increase in total word counts from 2008 to 2009, but no obvious change from 2009 onwards.

The next scatter plot shows the relationship between unique word counts and year.



From above, we can see a quick increase in unique word count from 2008 to 2009, and it continues to increase, but with a slower rate from 2009 onwards.

The following graphs shows the relationship between the processing time and the total word count.



We can see a linear relationship between processing time and total word count. With a R^2 value of 0.96, we can conclude that as total word count increases, the time taken to process also increases.