Adrian Gellert

Professor Maximillian Bender

CS231 Lecture A, Lab D

8 November 2022

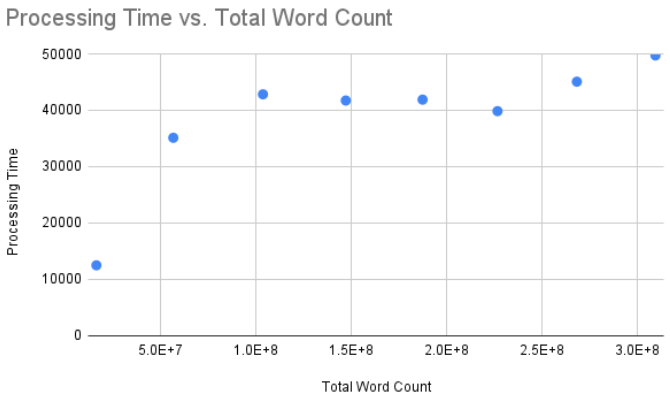
Adrian's Project 6: Reddit Comments Analysis (2008-2015)

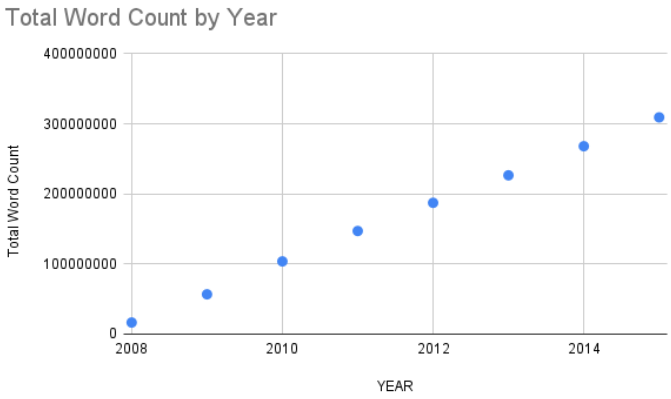
ABSTRACT

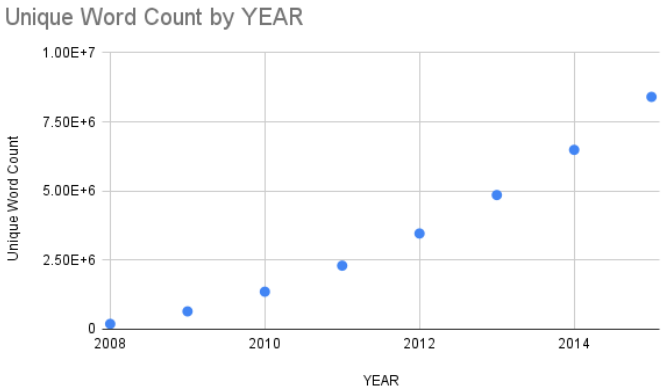
In this project we implemented a binary search tree to determine the frequency of words within text files efficiently. We fulfilled this goal, by using three classes. One class (MapSet) contained all the methods needed to construct a binary search tree. Another class (BSTMap) implemented MapSet for search operations to go through the binary search tree. Finally, we used a word count class to process the text files into a binary search tree and create a word-count file for analysis of text files.

Through analysis, we find that on average, the relationship between processing time and word count is logarithmic. We also find that the relationship between total word count or unique word count and year is a positive linear relationship, although unique word counts start out slightly exponential over time.

RESULTS


Processing Time vs. Total Word Count
50000
40000
30000
20000
10000
0
Processing Time
5.0E+7
1.0E+8
1.5E+8
2.0E+8
2.5E+8
3.0E+8
Total Word Count


Total Word Count by Year
400000000
300000000
200000000
100000000
0
Total Word Count
2008
2010
2012
2014
YEAR



REFLECTION

We used a binary search tree instead of other data structures because the method for searching for items within a binary search tree will always be shorter than searching for items within Arrays/Array Lists/Linked Lists. In contrast to these data structures, which have an O(n) search method, when searching for items within a BSTmap each step splits the number of nodes to search through by half, resulting in an O(log n) runtime complexity.

COLLABORATION

I got help from Catherine (“Jaime”) Yockey for code that we had similar errors that she had fixed. Bender helped me fix errors that resulted in the wrong output when testing my toString method. I also spoke with Claire while writing the report to understand the functionality of each class clearly.