

Thomas Wiles

Programming Assignment 5

Spellchecking with a BST Dictionary

May 1, 2017

Abstract: The goal is to spellcheck a text for grammatical accuracy. It is accomplished using a dictionary formed from an array of BinarySearchTrees using a text file dictionary.txt. Once the dictionary.txt file is read, each word can be loaded into the appropriate BinarySearchTree using the first letter of each word. Then, the text file oliver.txt must be spellchecked against the words in the dictionary. After each line is split into words, they can be compared to the elements of the BinarySearchTree dictionary. The results are that 889,754 words were found in the dictionary. It had to do an average of 16.11 comparisons to find them. The words not found in the dictionary numbered 17,888, and it had to do an average of 12.49 comparisons to determine they were not in the dictionary. Obviously this is significantly fewer comparisons than the LinkedList dictionary required. On average it did 3,500 comparisons to find a word and 8,700 comparisons to determine it wasn't in the dictionary. The reason for this enormous difference is that the BST sorted the words as they were entered. Every comparison cut the search space in half, meaning the words were found much more quickly. Thus BinarySearchTrees are a much more efficient way of storing large amounts of sortable data, where quick search methods are desired. LinkedLists have their advantages, but are not well suited for dictionary purposes.

run:

The number of words found is: 889754

The number of comparisons per word found is: 16.11

The number of words not found is: 17888

The number of comparisons per word not found is: 12.49

BUILD SUCCESSFUL (total time: 1 second)