CPSC 131 Fall 2018

Project 6: Word counting using a hash table

Introduction

This project is similar to Project 5 in that it analyzes words in English text but we will use a hash table to store words and their line numbers. The main task is to count word co-occurrences: the number of times two words occur together. Text analysis using word co-occurrences is an approach used in linguistics.

Objective

You are given a partial implementation of class TextAnalysis. The class has a single member variable, of class **std::unordered_map** which is the C++ Standard Library's implementation of a hash table, to store words (key) and all its line numbers in a vector (value). Your job is to complete three public member functions:

1. add_word(word, line): add word and line to the hash table
2. countWord(word): count the number of occurrences of the word
3. countTwoWords(word1, word2): count the number of lines which contain both word1 and word2. Note that if a word appears multiple times in a line, it should not be double counted. For example, if "hello" occurs five times in lines 2, 3, 3, 4, and 7; and "world" appears three times in lines 3, 4, and 4, then countTwoWords("hello", "world") should return 2.

The main program tests these methods using three separate pieces of text, all from the book: *Harry Potter and the Sorceror's Stone*. The first piece of text is the first paragraph from the book. The second piece is the first page (and a bit more to finish the paragraph). The last piece is the entire first chapter from the book.

Source Code Files

You are given "skeleton" code files with declarations that may be incomplete and without any implementation. Implement the code and ensure that all the tests in main.cpp pass successfully.

- TextAnalysis.h: This is to be completed
    - Note: the functions to read from a file and split a line into words is already given to you. You need to implement the remaining 3 public member functions.
- main.cpp: This tests the output of your functions.
- README.md: You must edit this file to include the name and CSUF email of each student in your group.

Hints

It is easiest to store **all** line numbers of a word as its value (vector) in the hash table. Then, the word count is just the size of the vector. But when counting word pairs, multiple occurrences of a word in a line should be handled.

Obtaining and submitting code

Click the assignment link to fork your own copy of the skeleton code to your PC.

https://classroom.github.com/g/sOrmUN3f

Development environment

The test platform is Linux with the g++ -std=c++14 compiler. For this reason, the recommended development platform is Linux.

Linux environment

To attempt to compile the test program, use the following command:

```
clang++ -g -std=c++14 main.cpp -o test
```

To attempt to run the compiled test program, use the following command:

```
./test
```

Grading rubric

Your grade will be comprised of two parts, *Form* and *Function*. *Function* refers to whether your code works properly as tested by the main function (80%). *Form* refers to the design, organization, and presentation of your code. An instructor will read your code and evaluate these aspects of your submission (20%).

Deadline

The project deadline is **December 14th at 11:55pm**.