CS4000

Homework # 7: Needle in a Haystack: Hidden Messages, Hadoop Compilation, Debugging, and Testing

due Friday, April 26th, 2019, 11:59 p.m.

(25 pts.)

Introduction

Email and other electronic media are often filtered for offensive language using fast, but relatively simple means (think finite automata). However, these simple filters are easily tricked. For example, if your filter was looking for the phrase “bad language,” you could trick it by inserting characters between the letters, e.g., by writing “..b..a..d...l..a..n..g..u..a..g..e...”. In this case, this text is easily recognizable by a human as the words “bad language”, but harder (but not impossible) to recognize by a simple filter.

For this assignment, you are looking for potential “secret messages” embedded into tweets (both real and simulated). In particular, you are trying to find which Twitter users are posting “secret messages” to their followers. You discovered that, given a string, such as “secretMESSAGE”, you can see whether those letters are embedded (in that order) in a tweet in $O(n)$ steps, where n is the number of characters in the tweet. You wrote a couple (2) of Hadoop programs to output the number of times each Twitter user tweeted a message that had the string “secretMESSAGE” embedded somewhere in the message. Unfortunately, your little brother Ike got into your account and messed up the program. The first program won’t compile. The second program runs, but it doesn’t get the right answer. You’ll have to fix them.

The Data Format

For this assignment, the Twitter data that you will be using has been pre-filtered to remove information about the Twitter user (`screen_name`) and the text of the tweet. Furthermore, tabs and newlines within the tweet have been converted to spaces. So, the data files for this assignment are provided in a line oriented format, where each line contains the screen name of a Twitter user and the text of tweet, where the screen name and the tweet are separated by a tab character `\t`.

Part I: Compiling Java Hadoop Programs (10 pts.)

Your first program `SecretMessage.java` looks for the fixed string “secretMESSAGE” in tweets. The provided program does not compile. It also has some warnings.

1. (3 pts.) Fix the compilation errors in the file `SecretMessage.java`. What lines did they occur on?

2. (3 pts.) Remove the compilation warnings in the file `SecretMessage.java`. What fixes did you make? Explain.

3. (3 pts.) Run the Hadoop program on the files `processed_tweets.txt` and `random_data.txt`. What is the output?

4. (1 pt.) How many blocks are used to store `random_data.txt` on the Hadoop Distributed File System?

Part II: Fixing a Broken Hadoop Program (15 pts)

The second program `GeneralMessage.java` is intended to be a general version of the first program. As the first commandline parameter, the user passes in a string that replaces the hard-coded “secretMESSAGE” in the first program. To run this second program, issue the command

```
yarn jar program.jar GeneralMessage secretMESSAGE Input Output
```

Here, “secretMESSAGE” could be anything.

The current program does not run correctly. Fix this program in such a way that it performs just like the original program when using the string “secretMESSAGE” as a commandline parameter, and so that it works on other strings as well.

1. (7 pts.) Fix the program `GeneralMessage.java` so that it compiles and runs correctly. What lines in the program did you fix? Be specific.

2. (4 pts.) Run the corrected program on the files `processed_tweets.txt` and `random_data.txt` and the string “ScaryMess”. How many users have tweets that match that string? Give one of them.

3. (4 pts.) Run the corrected program on the files `processed_tweets.txt` and `random_data.txt` and the string “SecretMessage”. How many users have tweets that match that string? Give one of them.