Eric Hunzeker & Brian Choromanski

CS 1699 - Final Project

Chatree Sangpachatanaruk

27 April 2019

Report

** For output screenshots on multi-word searches, see ‘report/search-results/’ **

Our application is a python3 command line interface which takes in ‘part-r-00000’ from mapreduce’s output and parses it into an inverted index data structure in python. When a user searches for a word or multiple words, the program will return the context of the three documents with the most occurances of each word. For example, if you type 3 words, the output will be nine contexts, one for each word for each top three documents.

We implemented the inverted index file using both mapreduce and spark. After this, we were able to handle the user input and searching in python. Also, if you run ‘driver.py’, it has to be from the tiny-google folder. Type ‘python3 src/driver.py’, because the folders that ‘part-r-00000’ and all of the docs are in are hard coded. Also, if you get the error “Traceback (most recent call last):

```
  File "src/driver.py", line 51, in <module>
    searchlines = f.readlines()
  File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/codecs.py", line 322, in decode
    (result, consumed) = self._buffer_decode(data, self.errors, final)
```

UnicodeDecodeError: 'utf-8' codec can't decode byte 0x93 in position 1142: invalid start byte", it is because the txt file starts with an empty line. If this happens, look at the last line of output and go to that file and delete the empty first line and it will work.

Overall, Spark was much easier to write and run for the most part. After figuring out how to get it running, spark was fast and required significantly less code. Mapreduce also was fast, but required three java files and much more code.