Peter Jough

IS392 Report

For this assignment, I first installed and imported the necessary libraries, afterwards I setup a variable, “seedUrl” & “seedUrl2” which has the seed URL topic, which in this case I wanted to be about oranges (the fruit).

For the related criteria based on the seed URL, I chose the following 10 terms and stored them in an array which I would later use to compare for similarity.

tenWords = ['fruit','species','health','location','farming','history','nutrition','biology','food']

These words relate to oranges, from it’s history, the science behind it, etc.

I then created a function “mainCheck” which takes in the seed URL, parses it then scans the webpage for all “a” tags. There’s a for loop in the function and it appends all the hrefs (the links) into a blank array called “compileURL,” which serves as our queue that will later be used.

In the second for loop within the function, we’re no comparing whether the seed URL has at least 2 matching terms that we established in the tenWords array to make sure the page has content I’m specifically looking for.

We have a counter variable, “counterN,” that is an integer with a value of 0 and for every term that the function scans and is one of the same terms I had in the tenWords array it would add a counter. After +2 on the counterN counter, we would print out a confirmation string text that the page has what I was looking for and gives the function the permission to write the information and store the webpage HTML code into the text file and breaks the loop. Within the function it displays the response type with “200” or “404” when it says 200 we know that the crawler is getting hits from the hyperlinks it’s scanning. When it’s 404 that means that particular hyperlink is dead. After it finally finds the 2 word criteria it prints a string so we know that hyperlink qualifies and a next text file is made with the associated information.

After that, the loop breaks and we move outside of the function and appending all the good urls into a new empty queue once it passes the test of having at least 2 credentials from the tenWords. The program is perhaps not the most efficient in run time since I’m using try & catch which I used to debug some earlier problems. As a result the program runs slows despite it running objectively fine.

Outside of the function, we run a while loops that will check our queue that has all the stored URL and once it’s > 501 websites it’ll stop. Outside of that while loop, a