Ahmed Alhaj
cmsc476
feb/19/2019
Report : phase1

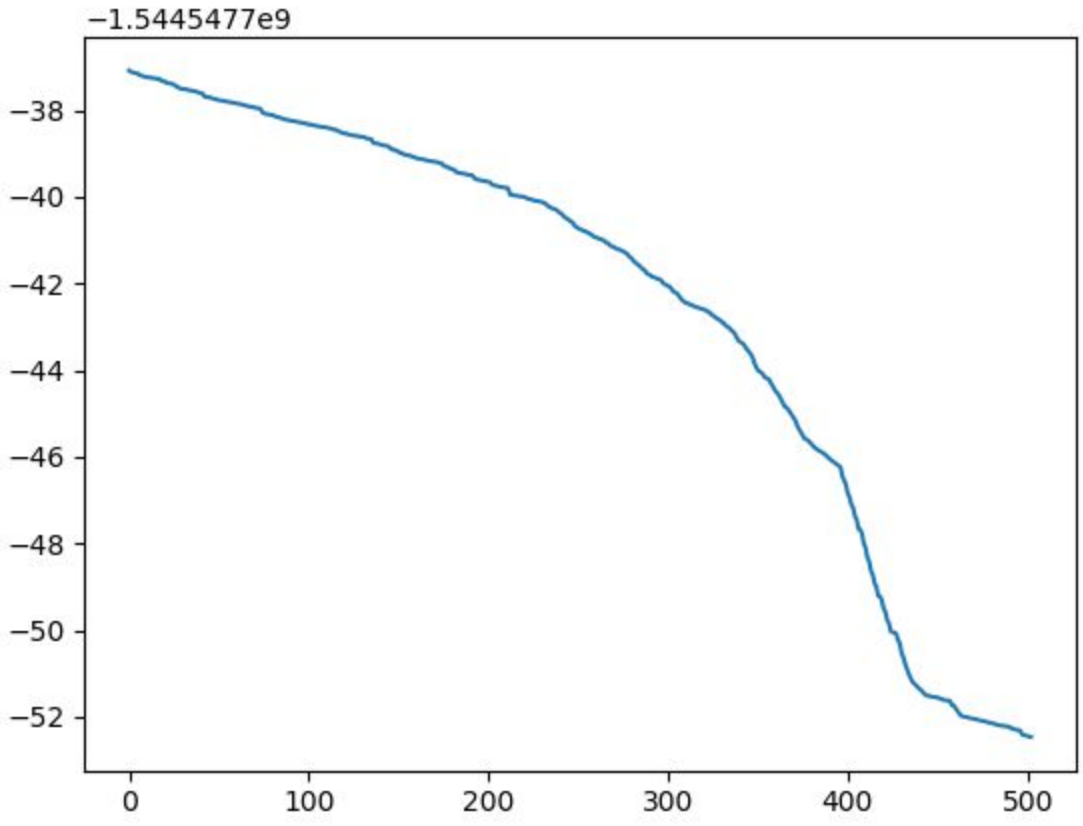
In this phase of the project, the task is to read around 500 html file, tokenized them and determine the frequency for each token and write it to a different file. There will be 2 output files one sorted by the frequency and the other sorted alphabetically by the token. So the first task is to view the files and identify some sort of pattern that can make it easy to process. The next goal is to make a plan as what is exactly needs to be done to read the file appropriately to get clean text. I decided to first get cleaned the file from the html code, clean unwanted characters and any misleading text.

The first task is how to get rid of the html code, I used html2text which was pretty easy, useful and efficient. The next step is getting rid of the unwanted characters in the hope that it would be easy to extract some useful information out of the files. The way I have done it is just by reading all file and filter them html syntax and concatenate them into one string. The last part which is getting the frequency was a bit Tedious, I attempted to write my own algorithm to get the frequency but it was not successful attempt. So I contacted the TA and he suggested using Counter and that what I ended up using. finally I sorted the files based on both the frequency and the text.

To address the comparison between the tokens, I did not have a classmate to share my output with. So I just attempted again using nltk for tekonization, the first approach I just split the string that contain the files and them pass to the Counter module to get the frequency. Which works pretty nicely. The second approach I am used the nltk tokenize and the FreqDist to get the frequency of each token. With regard to the comparison they were much difference with regard

to speed using The string split as tokenizer and counter for the frequency was little slower than using nltk tokenize and the Freq Dist. I also attempt to get the frequency of word with respect to each files which was a bit divergent from the assignment but I thought it would pretty useful.

This plot represent time consuming to process the file tokenize and get the frequency for using The tokenizer str::split and counter for the frequency



This plot represent the time consumed for the file to be processed using nltk tokenize and FreqDist() function, to get the frequency for each token