Methods II Homework 2

Tawa Suleman

8 March 2021

The first challenge I had with this homework assignment was figuring out where to start. I had initially misread the homework instructions, so I was struggling to correctly produce a code for a different task. I originally interpreted the task as having the user input what they wanted to have in the `train`, `dev` and `test` sets, rather than the user inputting the files in which they wanted the sets stored. Once I understood the task, it was much easier to get started.

However, as I tested my code, I received several errors. The first error I got was "Python not found". This error surprised me, as I had successfully run other codes during class. I realized, then, that it was because of the *shebang* at the beginning of the code. When calling python, my computer accepts the command `py` rather than `python`. So, I had to edit the shebang to `#!/usr/bin/env py`. After this I didn't receive the previous error, but my code still wasn't working. Now, it would run but would not terminate. I thought I had made an error somewhere in my code, so I opened a new code and started from scratch once more, being careful as I went line by line. When I still encountered the same error no matter how many times I rewrote the code, I was frustrated to say the least. I considered perhaps the size of the `conll2000.tag` file might have been slowing down the process, but even when I tested on shorter texts I encountered the same problem. After tinkering with the code a few times more, though, I rid my code of the shebang entirely and that seemed to do the trick.

When I tested the code on a smaller piece of text, I noticed that the dev set would come out empty since I used floor division in order to keep the list indexes integers. Since the text was so short, the floor division resulted in the index being 0, and the resulting list to be `corpus[6:6]`, an empty list. Though for a serious tool, one would probably be using larger corpuses and not encounter this problem, I decided to include an if statement that would help to handle this situation.

Another challenge I encountered was trying to compute the SHA-256 checksum. When using the command shown in the instructions, my computer did not recognize the `shasum` command. I figured this was another python vs py situation in which I would have to search and find the equivalent that my computer would accept. After searching online, the closest command I was able to find was `certutil -hashfile` (file name) `SHA256`. Using this to complete the third step of the assignment, I was able to see that the files still had the same SHA-256 checksum when I ran the script over the file again. However, this command did not allow me to input multiple files at once.