Maria Karamihaylova
Methods in Computational Linguistics II
HW 2
Due: March 8, 2021

I found this assignment extremely challenging but not impossible. I was able to learn a lot of new coding concepts through hours of trial and error as well as additional research. I initially had to re-watch the lecture and practicum videos in order to get a grasp of argparse. In addition, I watched several YouTube videos on argparse to deepen my understanding. I am still getting used to using git and VSCode so I had not realized I wouldn't be able to access the "conll2000.tag" file unless I was working on "split.py" out of "hw2-mkaramihaylova" folder. The most challenging part for me was figuring out how to write the data into the 3 file paths (training, development, test). Once I determined that I needed to write nested for loops to begin this task, I initially made the mistake of not indenting the for loop, resulting in a NameError because "corpus" was only defined within the "main" function. Once I indented the nested for loop, I struggled to convert my "word" element from a list to a string. First, I attempted to use "print(str(word))" but I was unsuccessful. I discovered that the "print(' '.join(word))" command was better suited to converting the lists into the desired string output. My challenge continued to be printing the string to a file. I tried "file=sink" in several ways but was unsuccessful. After some research, I added "file=open(train.txt, "a") to my print statement instead; this created a new dev.txt file while contained my sliced data in the form of strings. However, I then encountered another challenge because used the "wc -l" command-line tool, I received a message stating "No such file or directory." I suspected my error may have been in creating a ".txt" file so I replaced "train.txt" with "args.train" and ran the program again. However, I again received a "No such file or directory" message. I finally realized I had to output to a file called "train.tag." Ultimately, I was able to confirm that the file lengths were correct using the "wc -l" command line tool once more. The next step was to synthesize my three sets of nested loops into the "write_tags" function, which I nested under the "main" function, to avoid repetition in my code and to avoid hard-coding (as far as I understand it). My next step was to tackle randomization. I was easily able to use "random.shuffle" to randomize the corpus data but seeding the PRNG was tricky. Once I read the random documents, I was able to add a "--seed" flag and made it required. I confirmed that if the "--seed" flag is omitted, I received an error message. However, when I checked the "shasum" command line tool, I received different values. I decided that instead of "random.seed," I would try the "random.Random" method, but I continued to receive different values when checking the SHA-256. Unfortunately, I was unable to figure out this last bit of the task.