Code Sequence

Import needed libraries

- spacy
- os
- defaultdict, Counter
- csv
- re
- Tokenizer(**from** spacy.tokenizer **import** Tokenizer)

cleanText Function

- Have text as parameter
- Set the text to lowercase
- Use the pattern [^A-Za-z—\-\'\’\ ] and re library to replace every character that is not a letter, —, -, ', ’ or a whitespace with a whitespace
- Replace multiple whitespaces (the \s+ pattern) with just one whitespace
- Return the result

findTxts function

- Have path as a parameter
- Initialize an empty list
- Loop over the folders found in TextFiles (use os.walk() with the path parameter)
- Loop over the files found in the folder
- Check if they end in .txt (.endswith('.txt')
- Open the file and read from it
- Extend or append the cleaned text to the list
  list.extend(cleanedText(text))

Write inside your main

- Write if __name__ == "__main__":
- Initialize spacy with "en_core_web_sm“ and a max length of 1529140
- Initialize the tokenizer with the the vocab in spacy (https://spacy.io/api/tokenizer#_title there is an example at __init__)
- Initialize an empty list
- Run a for over the result of the tokenizer pipe with a batch size of 50
- Extend the empty list with the result of the pipe
- Outside the loop, create a Counter dictionary by provinding Counter with the list that contains the results

Saving the results(still inside the main)

- Use the with syntax to open a file with parameters "Counts/CSRWordFreqDict.csv“, mode="w+", newline="", encoding=' utf-8’ (Make sure to have created the Counts folder)
- Write the headers of the csv 'word', 'count’.
- Use the most_common() method on the Counter dictionary to get a list of sorted tokens. It will return the id number of the word and its count in a tuple.
- For loop over that list(remember that it contains tuples) and write them to the csv. To get the actual word use tokens.vocab.strings[id number of the word]