CITS1401 Project 2 Pseudocode

Semester 2, 2019

Main function

- Inputs: textfile1, textfile2, feature
- Check that the first and second argument are files that exist in the directory. Print an error if they don't exist
- Run the appropriate function depending on the feature provided as input, such as:
    - Run the conjunctions() function if the specified feature is 'conjunctions'
    - Run the unigrams() function if the specified feature is 'unigrams'
    - Run the punctuation() function if the specified feature is 'punctuation'
    - Run the composite() function if the specified feature is 'composite'

Generate_Lists function

- Inputs: textfile1, textfile2
- Check that the two inputs are files that exist in the directory. Print an error message if they don't
- Create two lists to hold the data from each text file - list_file1 and list_file2
- For both textfile1 and textfile2:
    - Open the files in 'read' mode
    - Run a for-loop that iterates through each line
    - Run a nested for-loop that iterates over and separates each word individually using the split() function
    - Append each word to the appropriate list for both text file's – list_file1 and list_file2
- Return list_file1 and list_file2 for use by later functions

Conjunctions function

- Inputs: textfile1, textfile2
- Call the generate_lists function to capture the data (lists) returned into two variables called document1 and document2
- Create two new lists to hold the matching conjunction words contained in the document1 and document2 lists – called doc1_conjunction and doc2_conjunction respectively
- Create a list called conjunction_list that holds strings for all conjunction-type words
- Run a for-loop over the conjunction list
    - Run a nested for-loop over the document1 and document2 lists
    - Convert the words in each document to lower-case to allow for better comparison
    - If the words match, append them to the appropriate doc_conjunction list
- Initialize two dictionaries: profile_conjunctions_count1 and profile_conjunctions_count2
- Iterate over the words in doc1_conjunction and doc2_conjunction, and if that word is in the appropriate dictionary for that list, increment the count of that particular word by 1, otherwise leave the count unchanged

- Calculate the distance by calculating the difference in word counts for matching words between the two profiles, then finding the Square Root of that total difference
- Return the calculated distance, followed by the two completed dictionary profiles

Unigrams function

- Inputs: textfile1, textfile2
- Call the generate_lists function to capture the data (lists) returned into two variables called document1 and document2
- Initialize two dictionaries: profile_count_unigrams1 and profile_count_unigrams2
- Run a for-loop on the document1 and document2 lists, iterating over the words
    - Convert the words in each document list to lower-case to allow for better comparison
    - if the word exists in the appropriate dictionary for that list, increment the count of that particular word by 1, otherwise leave the count unchanged
- Calculate the distance by calculating the difference in word counts for matching words between the two profiles, then finding the Square Root of that total difference
- Return the calculated distance, followed by the two completed dictionary profiles

Punctuation function

- Inputs: textfile1, textfile2
- Call the generate_lists function to capture the data (lists) returned into two variables called document1 and document2
- Initialize a list holding the punctuation characters
- Initialize two dictionaries: profile_count_punctuation1 and profile_count_punctuation2
- Run a for-loop iterating over the document1 and document2 lists
- If the document1 and document2 lists contain the punctuation characters, update the respective dictionaries with the count
- For certain characters, only update their count if they are surrounded by letters. Do this by using indexing to check the characters before and after the punctuation characters. i.e if [-1] and [+1] is equal to a letter
- If punctuation such as a full stop is not at the end of a sentence (is not followed immediately by a blank space), convert it to white space such as “ “.
- Calculate the distance by calculating the difference in word counts for matching words between the two profiles, then finding the Square Root of that total difference
- Return the calculated distance, followed by the two completed dictionary profiles

Composite function

- Inputs: textfile1, textfile2
- Check that the two inputs are files that exist in the directory. Print an error message if they don’t
- Repeat the same steps as in the conjunctions() and punctuation() functions to count the occurrences of punctuation characters and conjunction characters

- Create two lists to hold the data from each text file - list_file1 and list_file2
- For both textfile1 and textfile2:
    - Open the files in ‘read’ mode
    - Run a for-loop that iterates through each line in both text file’s
    - Run a nested for-loop that iterates over the text and separates the text into lists of paragraphs by splitting where a new-line character is encountered.
    - Run a nested for-loop that iterates over each paragraph and separates each sentence using the split(“. “) function, where you are splitting based on the end of a sentence
    - Now that the text is split into lists of paragraphs, and then lists of sentences, count the number of words in each sentence
    - Calculate the average words-per-sentence by taking the count of words and dividing the total number of words by the number of sentences
    - Calculate the average number of sentences per paragraph by taking the count of sentences and dividing the number of sentences by the number of paragraphs
    - Append the dictionaries with the newly calculated averages for each text

- Calculate the distance by calculating the difference in word counts for matching words between the two profiles, then finding the Square Root of that total difference
- Return the calculated distance, followed by the two completed dictionary profiles