# Psuedocode for project 2 CITS1402

# by Luke Bray 22754415

1. Function main takes in textFile1Path, textFile2Path and ProfileMethod:
2. Check textFile1Path and textFile2Path are valid
3. Read in textFile1Path as string
4. Read in textfile2Path as string
5. For each character in textFile1Path:
6. If character is an escape character, delete it
7. If the character is “-“ and the character after is “-“:
8. Replace the two characters with a space
10. If the feature is “punctuation”:
11. Call punctuationProfileCalculator with textFile1Data
12. Call punctuationProfileCalculator with textFile2Data
13. If the feature is “unigrams”:
14. Call unigramProfileCalculator with textFile1Data
15. Call unigramProfileCalculator with textFile2Data
16. If the feature is “conjunctions”:
17. Call conjunctionProfileCalculator with textFile1Data
18. Call conjunctionProfileCalculator with textFile2Data
19. If the feature is “composite”:
20. Call compositeProfileCalculator with textFile1Data
21. Call compositeProfileCalculator with textFile2Data
22. Else:
23. Exit the program
25. Call calculateScore with calculatedProfile1 and calculatedProfile2
26. Output calculatedScore
28. Function punctuationProfileCalculator takes in textData:
29. Define punctuationDict = {“;” : 0, “,” : 0, “-“: 0, “ ‘ “ : 0}
30. For character in textData:
31. If character = “;” then punctuationDict[“;”] += 1
32. If character = “,” then punctuationDict[“,”] += 1
33. If character = “-“:
34. If the character before and after is a letter:
35. punctuationDict[“-“] += 1
36. If character = “ ‘ “: #the space in the string is just to make it easier to read
37. If the character before and after is a letter:
38. punctuationDict[“ ‘ “] += 1
39. Return punctuationDict
41. #pseudocode continues on next page
42. Function unigramProfileCalculator takes in textData:
43. Set unigramProfile to blank dictionary
44. SplitTextData = split textData on space characters
45. For word in splitTextData:
46. unigramProfile[word] += 1
47. Return(unigramProfile)
49. Function conjunctionProfileCalculator takes in textData:
50. Set conjunctionProfile to blank dictionary
51. ValidConjunctions = ["also", "although", "and", "as", "because", "before", "but", "for", "if", "nor", "of", "or", "since", "that", "though", "until", "when", "whenever", "whereas", "which", "while", "yet"]
53. For word in (split textData on space characters):
54. If word is in ValidConjunctions:
55. conjunctionProfile[word] += 1
56. Return conjunctionProfile
58. Function CompositeProfile takes in textData:
59. PunctuationProfile = call punctuationProfileCalculator with textData
60. conjunctionProfile = call conjunctionProfileCalculator with textData
61. For item in conjunctionProfile:
62. punctuationProfile[item] = conjunctionProfile[item]
63. Return(punctuationProfile)
65. Function CalculateScore takes in profile1 and profile2:
66. sumOfDifferences = 0
67. For item in profile1:
68. sumOfDifferences += (profile1[item] – profile2[item]) ^2
69. Score = squareroot(sumOfDifferences)
70. Return(score)