CS366 Final Assignment Tests

For my solution to work correctly you might have to change the global variable. It is currently, training_folder = '/Users/[Username]/ Documents/CS366(Internet of Things)/FinalAssign/training/'
This is just the local folder for my Pycharm IDE to make it easy on myself.
(Note to professor) Change it back to training_folder = './data/training/' so it works on your rasp pi.

I really enjoyed this assignment, it was pretty challenging and there are definitely more implementations I can do to my solution. The first thing I changed is the removal of punctuation marks function that you had. Here is my function: re.sub(r'[^\w\s]','',s) This seemed to work better for me in lab3 and in this instance.

The second function I changed is, all_file_list() so that it contains the folder "Tech" for more variety when looking at training data.

With this assignment I was definitely confused in the beginning. None of my solutions would work and I was confused at the answers I was getting. I took some time to print out and understand all of the lists in the test function. One big problem I found was that the dist_list already had the 'tag' and 'distance' for each training article compared to the input txt. Then for some reason there was a vote_result for loop that only took into account the number of articles not the frequency of words in each article. For this assignment, the number of articles in each training category is not important. What is important is how closely related the input txt is to all of the training data. I feel like my solution accurately represents the idea behind this assignment. Each article is compared against all articles in the training data. The training data article that is most closely related to the input article has the lowest distance vector. That will be evident in my tests when I input an article that is the same as one in the training data.

Covid Article Example:
Which article would you like to test? CovidExample.txt

How related your article is to the current training data (0.0 is identical):
38.27531841800928
50.57667446560717
38.742741255621034
50.40833264451424
74.06078584514209
41.42463035441596
37.255872020394314
44.30575583375144
43.78355855797927
41.42463035441596
40.755367744629666
71.74956445860839
51.34199061197374
41.66533331199932
60.687725282795036
44.82186966202994

Total number of articles used for comparisons: {'Minnesota': 5, 'Health': 5, 'Tech': 6}
Your article's category according to our training data is: **Health**

Tech Article Example:
Which article would you like to test? TechExample.txt

How related your article is to the current training data (0.0 is identical):
38.1051177665153
36.64696440361739
27.09243436828813
35.32704346531139
66.69332800213226
31.827660925679098
30.610455730027933
42.82522621072771
30.62678566222711
31.192947920964443
30.789608636681304
68.5638388656878
40.95119045888654
22.11334438749598
51.96152422706632

30.72458299147443

Total number of articles used for comparisons: {'Minnesota': 5, 'Health': 5, 'Tech': 6}
Your article's category according to our training data is: **Tech**

Minnesota Article Example:
Which article would you like to test? Minnesota.txt

How related your article is to the current training data (0.0 is identical):
42.1070065428546
53.68426212587819
56.70097000933934
50.00999900019995
86.06392972668631
52.03844732503075
57.532599454570104
55.235857918565905
62.56996084384263
51.048996072400875
62.10475022089695
54.55272678794342
48.92851929090027
54.91812087098393
50.941142507800116
57.55866572463264

Total number of articles used for comparisons: {'Minnesota': 5, 'Health': 5, 'Tech': 6}
Your article's category according to our training data is: **Minnesota**

You'll notice here that given an identical article to one of the training data articles, it returns a distance vector value of 0.0. This means that the article is identical to one of the training articles thus enforcing the idea that the distance vector data is correct.

Article Identical to one of the training data:
Which article would you like to test? IdenticalHealth.txt

How related your article is to the current training data (0.0 is identical):
46.968074263269514

48.8978527135906
45.27692569068709
48.53864439804639
80.2869852466762
44.50842616853577
46.03259714593562
0.0 <————
47.138094997570704
44.41846462902562
49.29503017546495
67.54998149518622
52.10566188045211
46.72258554489466
57.67148342118486
44.98888751680797

Total number of articles used for comparisons: {'Minnesota': 5, 'Health': 5, 'Tech': 6}
Your article's category according to our training data is: **Health**

Adding more articles to my training data will only increase the efficiency and accuracy of my solution. New articles added to the training data **must** be filtered such that they contain mostly information from their specific section. This is why companies like Netflix make sure to define each movie or title with a specific designation.

My solution however does not currently output if an article can be contained in two different categories. This is possible to be done but I didn’t have enough time to explore this idea.