NAME: OYEDAYO OYELOWO. Phone Number: +358469551643.

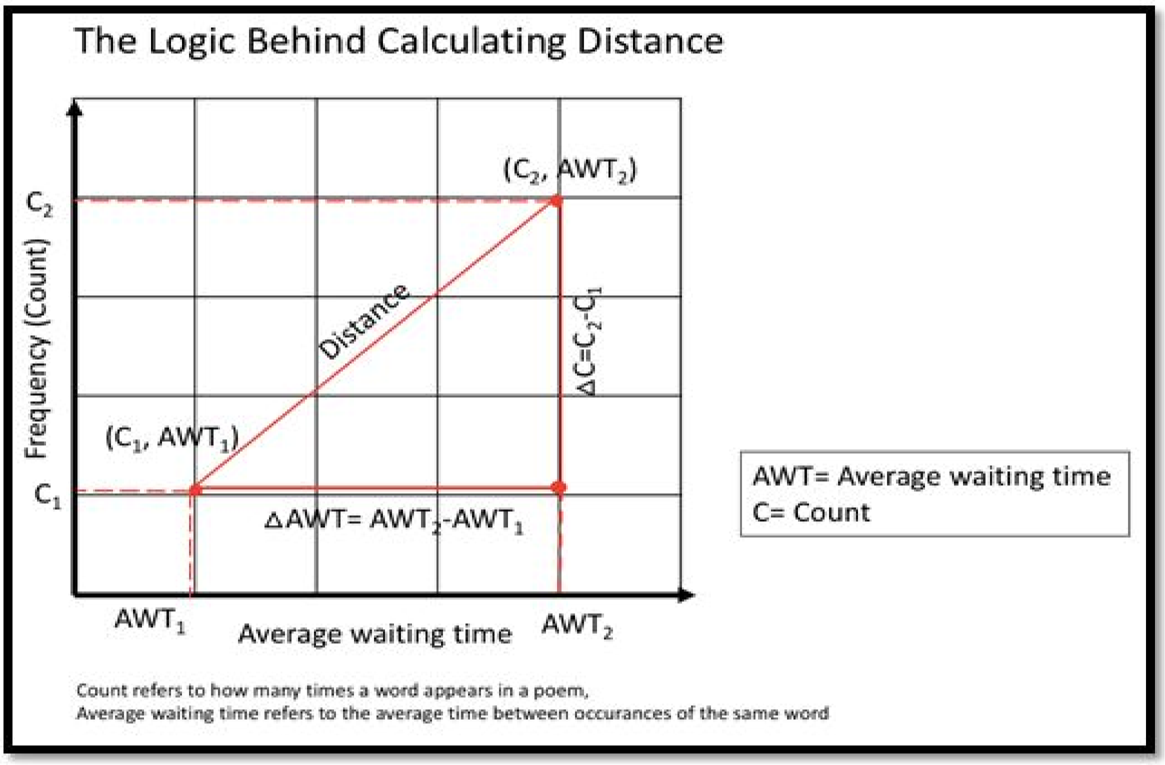
EMAIL: oyelowo.oyedayo@helsinki.fi, oyedayooyelowo@gmail.com

CALCULATING DISTANCE BETWEEN TWO POEMS

2. Additional question: Give a short description (do not implement) of an approach to calculate a distance (scalar) between two poems e.g. utilizing the stats you calculated above.

SOLUTION

The scalar distance between two poems can be calculated, using for example: the ***count(frequency)*** or by combining "***Count***" and "***Average Waiting Time***" generated above. Using count alone is simple but not as comprehensive as combining it with another criterium. Therefore, I would be adopting the ***Euclidean Distance Measure***. This gives an idea of the similarity between poems. "0" distance means that they are the same, the more the distance, the more dissimilar they are.



Count refers to how many times a word appears in a poem,
Average waiting time refers to the average time between occurances of the same word

Recall that scalar quantity(in our case; distance) has a magnitude/size but not a direction. The diagram above is a representation of how I would approach this problem because it has more distance criteria.

Firstly, I would tokenize the two poems. This means removing the stop words(which are words that do not add significant meaning to a sentence). This can be achieved with python libraries like "`sklearn`" and "`nltk`". It is also important to convert them to lowercase, in the process.

Secondly, we need to know the counts and average waiting time for each word. For example:

For each word First Poem = (C_1, AWT_1) *nth word*

For each word in second Poem = (C_2, AWT_2) *nth word*

Therefore, distance between them would be:

Distance(S) = $\sqrt{\sum (x2\ - x1)^2\ +\ (y2 - y1)^2}$, SideNote: following the pythagoras principle.

Distance(S) = $\sqrt{\sum (\ (AWT2\ - AWT1)^2\ +\ (C2 - C1)^2)\ )nth\ word}$

IMPLEMENTING IN PYTHON:

There are many off-the-shelf libraries that can be used for automatically calculating this distance such as "euclidean_distances" in "**sklearn.metrics.pairwise**", which can be imported as thus: "**from sklearn.metrics.pairwise import** euclidean_distances.

However, the procedures are:

1. Insert all the words in both poems in an array.
2. Append the frequencies(i.e count) of each word in first poem in another array.
3. Append the frequencies(i.e count) of each word in second poem in another array.
4. Calculate the the square of the frequency difference of every word(i.e step 3 - step 2)^2.
5. Append the Average waiting time of each word in first poem in another array.
6. Append the average waiting time of each word in second poem in another array.
7. Calculate the the square of the Average waiting time difference of every word (i.e step 6 - step 5)^2.
8. Sum step 7 and step 4 for each word.
9. Now, sum all the words above.
10. lastly, find their square root, to get the distance.

Example:

Words	C 1	C2	$(C2 - C1)^2$	AWT1	AWT2	$(AWT2\text{-}AWT1)^2$	$(C2 - C1)^2 +$ $(AWT2\text{-}AWT1)^2$
finland	8	2	4	20	10	100	104
is	0	6	36	0	5	25	61
very	7	0	49	8	0	64	113
beautiful	5	5	0	12	7	25	25

sum = 303

Distance = $\sqrt{303}$ =17.41