Document Distance

For the given two text documents, which could be essays, emails, or perhaps even code, document distance is a measure to determine how similar or dissimilar the two documents are. One of the applications of document distance is plagiarism detection. When two documents are very similar to each other they are marked as a suspect for plagiarism. The computation of document distance has the following simple steps.

1. Read the text document into a string
2. Make a words list and clean up the words
    a. Convert the given text into lowercase
    b. Split the lowercase text into words
    c. Remove any spaces or newline characters in the words
        i. As a result of splitting the text into words at times extra spaces and/or newline characters are present at the beginning or at end of the words
        ii. You can use the strip method of the string in python to remove such trailing spaces and newline characters.
    d. Remove non alphabet characters from each word
3. Remove stop words
    a. Stop words are very common words in the text such as a, an, the, etc. These words are removed as they are not useful for computing the document distance.
    b. Stop words are provided in a separate file. Load these words into a list or dictionary.
    c. Remove the stop words present in the words list created in the previous step.
4. Compute word frequencies for the two text documents.
    a. Iterate through the two words list and build a common dictionary
    b. The words in the words lists are keys in the dictionary
    c. For each word in the dictionary compute the frequency of the word in each document
5. Compute the distance using the cosine similarity formula given below.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Example Computation

D1 = “to be or not to be - William Shakespeare”
D2 = “to be or not to be - Charlie Rose”
Word frequencies - {to:[2,2], be:[2,2], or:[1,1], not:[1,1], william:[1,0], shakespeare:[1,0], charlie:[0,1], rose:[0,1]}
Numerator = sum(2 * 2, 2 * 2, 1 * 1, 1 * 1, 1 * 0, 1 * 0, 0 * 1, 0 * 1)
Denominator = sqrt(sum(2^2, 2^2, 1^2, 1^2, 1^2, 1^2, 0^2, 0^2)) * sqrt(sum(2^2, 2^2, 1^2, 1^2, 1^2, 1^2, 0^2, 0^2))