Document similarity will consist of three fundamental steps:

Split the documents in words. Compute the word frequencies. Calculate the dot product of the document vectors. **bold text**

```
import math
import string
import sys

# reading the text file
# This functio will return a
# list of the lines of text
# in the file.
def read_file(filename):

  try:
    with open(filename, 'r') as f:
      data = f.read()
    return data

  except IOError:
    print("Error opening or reading input file: ", filename)
    sys.exit()

# splitting the text lines into words
# translation table is a global variable
# mapping upper case to lower case and
# punctuation to spaces
translation_table = str.maketrans(string.punctuation+string.ascii_uppercase,
                  " "*len(string.punctuation)+string.ascii_lowercase)

# returns a list of the words
# in the file
def get_words_from_line_list(text):

  text = text.translate(translation_table)
  word_list = text.split()

  return word_list

```

Now that we have the word list, we will now calculate the frequency of occurrences of the words.

```
# counts frequency of each word
# returns a dictionary which maps
# the words to their frequency.
def count_frequency(word_list):

  D = {}

```

```
  for new_word in word_list:

    if new_word in D:
      D[new_word] = D[new_word] + 1

    else:
      D[new_word] = 1

  return D

# returns dictionary of (word, frequency)
# pairs from the previous dictionary.
def word_frequencies_for_file(filename):

  line_list = read_file(filename)
  word_list = get_words_from_line_list(line_list)
  freq_mapping = count_frequency(word_list)

  print("File", filename, ":", )
  print(len(line_list), "lines, ", )
  print(len(word_list), "words, ", )
  print(len(freq_mapping), "distinct words")

  return freq_mapping

```

Lastly, we will calculate the dot product to give the document distance.

```
# returns the dot product of two documents
def dotProduct(D1, D2):
  Sum = 0.0

  for key in D1:

    if key in D2:
      Sum += (D1[key] * D2[key])

  return Sum

# returns the angle in radians
# between document vectors
def vector_angle(D1, D2):
  numerator = dotProduct(D1, D2)
  denominator = math.sqrt(dotProduct(D1, D1)*dotProduct(D2, D2))

  return math.acos(numerator / denominator)

```

```
def documentSimilarity(filename_1, filename_2):


  sorted_word_list_1 = word_frequencies_for_file(filename_1)
  sorted_word_list_2 = word_frequencies_for_file(filename_2)
```

```
    sorted_word_list_2 = word_frequencies_for_file(filename_2)
    distance = vector_angle(sorted_word_list_1, sorted_word_list_2)

    print("The distance between the documents is: % 0.6f (radians)"% distance)


# Driver code
documentSimilarity('alice.txt', 'alicemodified.txt')
```

```
File alice.txt :
1326 lines,
257 words,
138 distinct words
File alicemodified.txt :
1335 lines,
259 words,
139 distinct words
The distance between the documents is:  0.042916 (radians)
```