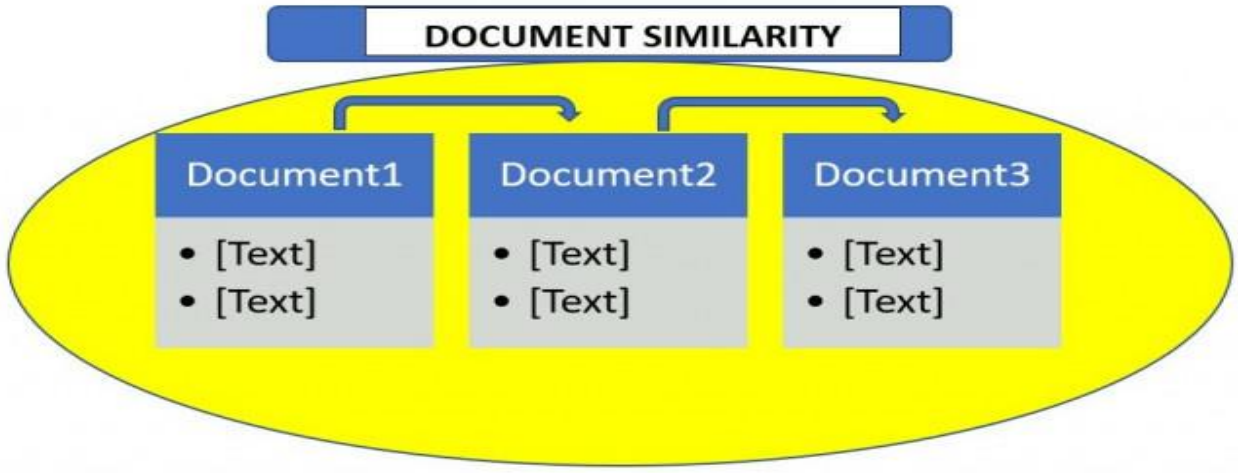
Assignment : Plagiarism Checker

KAPIL VERMA
ENTRY NO-2018CS10348

About My Implementation:

In this assignment, we have to build a Plagiarism checker or we can say document similarity checker. Plagiarism checker means, we take two txt file and check how much similarity they contain to each other's.



I have implemented this using cosine distance and cosine similarity algorithms. As this algorithm works on the words, I mean it will tell us how much file is similar based on common word so it's kind of syntactic algo nor a semantic algo.

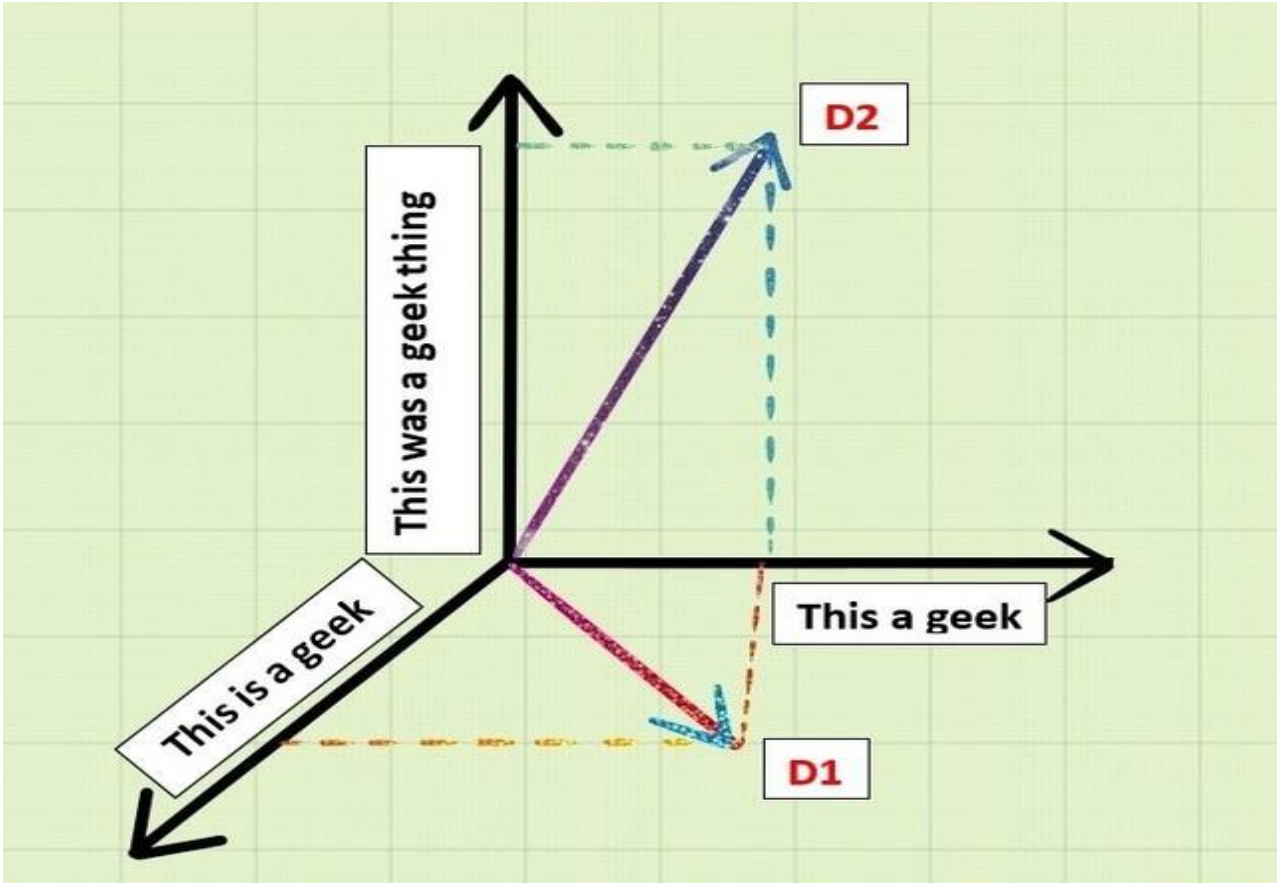
Cosine similarity and cosine distance algorithm:

Say that we are given two documents **D1** and **D2** as:

D1: “This is a geek”
D2: “This was a geek thing”

Both the document contains similar word:

D3: "This is geek"



Now if take dot product of two vector:

D1.**D2** = "This"."This"+"is"."was"+"a"."a"+"geek"."geek"+"thing".0

D1.**D2** = 1+0+1+1+0

D1.D2 = 3

As we know how to calculate dot product of two vector, Now we can calculate the angle between them:

Cos(d): |D1.D2|/|D1||D2|

2

Here d denotes distance between two documents which ranges from 0 to 90 degree. If d is 0 it means both the documents are exactly identical and if d is 90 then both documents are exactly non-identical.

Assumptions:

- Total of words should be less than 10,100.
- The length of words should be less than 100.
- The length of path of corpus file or test file should be less than 100.

How exactly are we using this algo:

So, what are we doing is that we taking two txt file and start reading one by one each of them. During reading txt file we store the words in the array of string and there is also array of int which store the frequency of word in that file. Like index i in string array would have a word and same index i in int array will represent that how many times we encountered that word in whole file.

After reading properly two files, we have two vector (int vector) having frequencies of every words. Now we can simply apply cosine similarity and cosine distance theorem.

Now we will do some calculation to find the angle between two vector and formula is

Cos(d): |D1.D2|/|D1||D2|

Functions:

Here is the list of important functions:

- **Int main(int argc, char* argv[]):**

This is most important function in c, as C program cannot run without this. We are taking the path of test file and corpus_files so this will be like:

Agv[1] = Path of test_file

3

Argv[2] = Path of corpus_files.

Now we have name of both file and folder, now we start reading them.

- **Int reading(char* testfile, char* corpus_files, struct datafile *tf):**

  Here we start reading both the inputs. Testfile input denotes path of testfile and corpus_files string is for path of corpus_files. First of all, we read testfile and store the data in the array of struct data. For reading testfile I called establise function where read testfile and store everything in tf. After this we start reading corpus_files directory and take file one by one. For reading txt file we use executing function.

- **Double executing(FILE* file, struct datafile *tf):**

Here we are reading file of corpus_files and already we have tf that stores all the information regarding testfile. We do some filteration during reading the txt file. After reading the whole txt file, we go for finding angle between the vector that is one for testfile and one for txt file that comes from corpus_files.

- **Double vector_angle(struct datafile *df, struct datafile *tf, int len2):**

  This function calculate the angle between two vector by c doing some mathematics and scale it in 100 to make sure we can represent it in percentage.

Filtration:

{

"the", "of", "and", "a", "to", "in", "is", "you",

"that", "it", "he", "was", "for", "are", "as", "my"

"with", "his", "they", "i", "at", "be", "this", "have",

"from", "or", "one", "had", "by", "but", "were", "all",

4

"we", "your", "when", "can", "there", "an", "which", "she",

"do", "their", "if", "will", "up", "then", "these", "so",

"her", "his", "would", "him", "into", "has", "could", "may",

"get", "did", "its", "been", "than", "s"

};

During reading we usually do some filtration like removing some words. As we know there are some words which comes more frequent in English literature, so by removing them in calculation we can have more accurate answer. Same thing I have done in this assignment we have made list of some words which comes more often, and we don't consider them during calculation.

Result:

Finally after calculating angle between the two vectors that is denoting frequency of words, we are printing result in percentage.

5