# Assignment 7 Report

DIWAKAR PRAJAPATI
2018CS10330

**AIM**: Plagiarism Checker for test files.

**Design Detail:**
I have used **Cosine** metric to find the similarity of two text files.
We assume a text file a N-dimensional vector of distinct words, where each dimension represents a word and the magnitude of the vector component is the frequency of that particular word.

Step 1: Read the corpus files and pre-process it.
Step 2: Vectorize the corpus files.
Step 3: Read the target file and pre-process it.
Step 4: Vectorise the target file.
Step 5: Use the vector to compute the cosine probability.

**Pre-Processing**
I : Scan the text and remove extra space, special characters including '.', ',' , '!', '?' , '-', ':', ';', '\"', '\'', etc.
II : Convert all the characters to lowercase to remove case sensitivity.

**Vectorisation**: Initially my vector is empty,
I : Take a word from the vector
II: Check if this word is already present in the vector
    Yes: Increment the frequency of that word.
    No: Add it to the vector
III: Move to next word until end of file.

**Cosine Probability**:
Each word in vector 1 is searched in vector 2, if a match is found then their freq is multiplied. Otherwise their multiplication becomes zero.
**While comparing words, I have handled to some extent the similarity of plurals and past/present/future forms of verbs,** by comparing the initial characters of the words only.
So even if I compare same text file, I may not give me 100% match , since the plurals, and verbs form would also be considered.

**Cosine(v1,v2) = v1.v2 / (sqrt(|v1|)* sqrt(|v2|))**, v1.v2 is the dot product and |v| is the square of magnitude.

**Time Complexity**
Obviously the time complexity grows and the corpus dataset size increase, so I would consider comparing two files and then multiply by the number of corpus files.
Assume we have
w1: size of corpus file
w2: size of test file

**Preprocessing them takes O(w1) + O(w2),**
Tokenization(for one file):
Each word in taken and then search in the vector, so if there are n number of words in file then time complexity of tokenization: $O(n^2)$

So if there are n1 and n2 number of words in corpus file and target file the,Time Complexity = $O(n1^2) + O(n2^2)$

Finding Cosine probability: Each word in vector 1 is searched in vector 2, if a match is found then their freq is multiplied.

So Time Complexity (n1*n2).

**Overall Time complexity with single corpus file**:
Let n1=number of words in corpus file
Let n2 = number of words in target file

Time complexity = $O(n1^2) + O(n2^2) + O(n1^2) + O(n2^2) + O(n1*n2)$
$= O(n1^2) + O(n2^2) + O(n1*n2)$

Space Complexity = O(no of character in corpus file + no of character in target file)

**Final Complexity with entire corpus data set**
**Time: $O(k.n1^2) + O(n2^2) + O(k.n1*n2)$ ,**
k = no of corpus files,
n1=avg size of each corpus file,
n2 = size of target file

**Space: O(total corpus dataset size + target file size)**