# Design Document for Plagiarism Checker

## Team:

- Chandrahas Aroori (f2016A7PS0100H)
- Naren Surampudi (f2016AAPS0206H)
- Aditya Srikanth (f2016A7PS0091H)

## Topic

Implementation of basic plagiarism checker using Python3.

## Introduction

**Plagiarism detection** is the process of locating instances of plagiarism within a work or document. The widespread use of computers and the advent of the Internet has made it easier to plagiarize the work of others. Most cases of plagiarism are found in academia, where documents are typically essays or reports. Our plagiarism checker checks the level of plagiarism between a given document and a corpus of documents.

## Libraries used

The python libraries used were:

1. Nltk: This library is generally used for Natural Language Processing, we have used it to tokenize the data and remove the stop words.
2. os:  This library is generally used for System Commands, we have used it for get the current working directory.
3. pickle: This library is generally used for serializing objects in python, we used it to serialize dictionaries which stored our data.
4. Math: This library is generally used for mathematical functions in python, we used for the logarithmic function.

## Implementation

### Corpus Preprocessing

The tf-idf score was calculated   was then calculated for each pair of (term, document) using the formula:

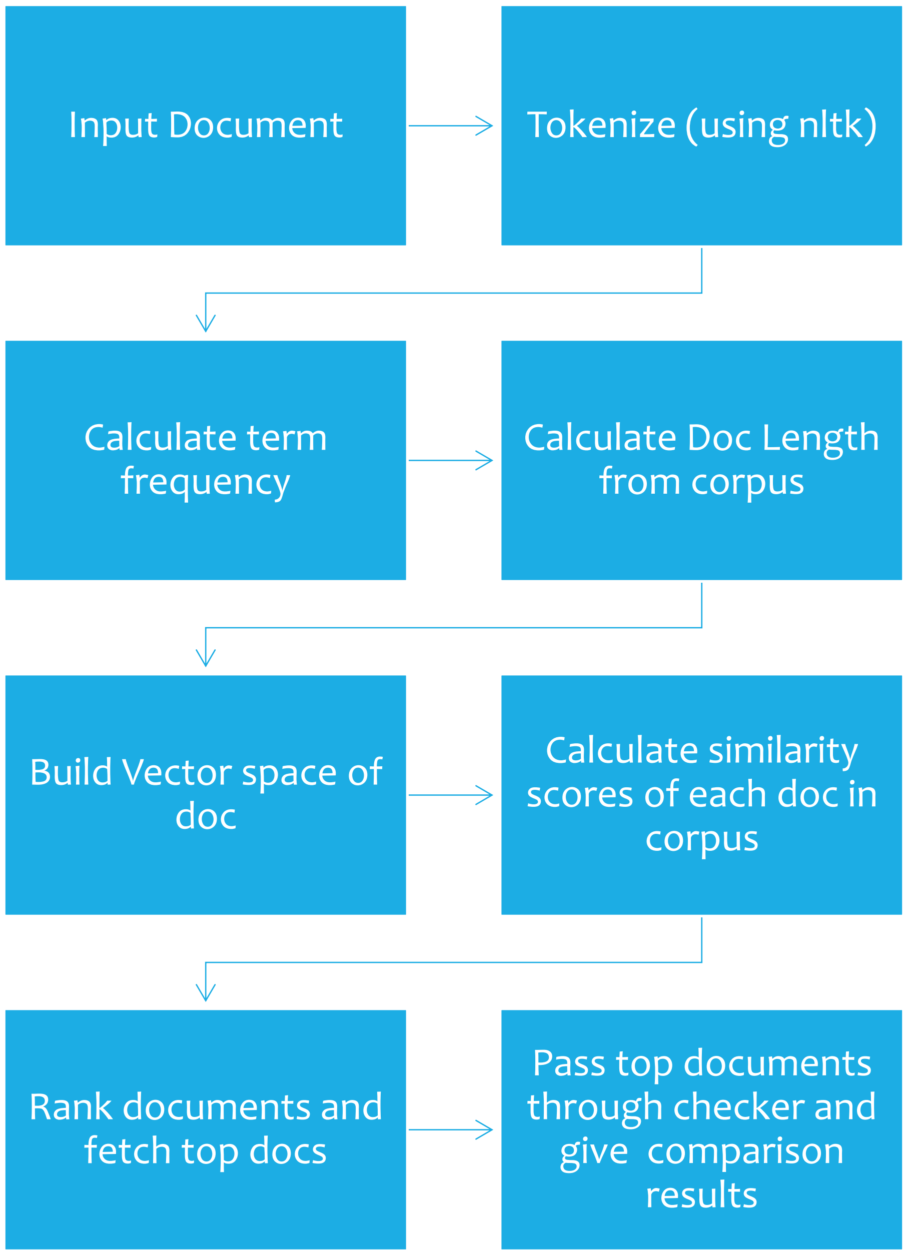
$$Tfidf = tf * idf \qquad\qquad where\ idf\ = \log(\frac{N}{df})$$

### Pseudocode for Tfidf calculation:

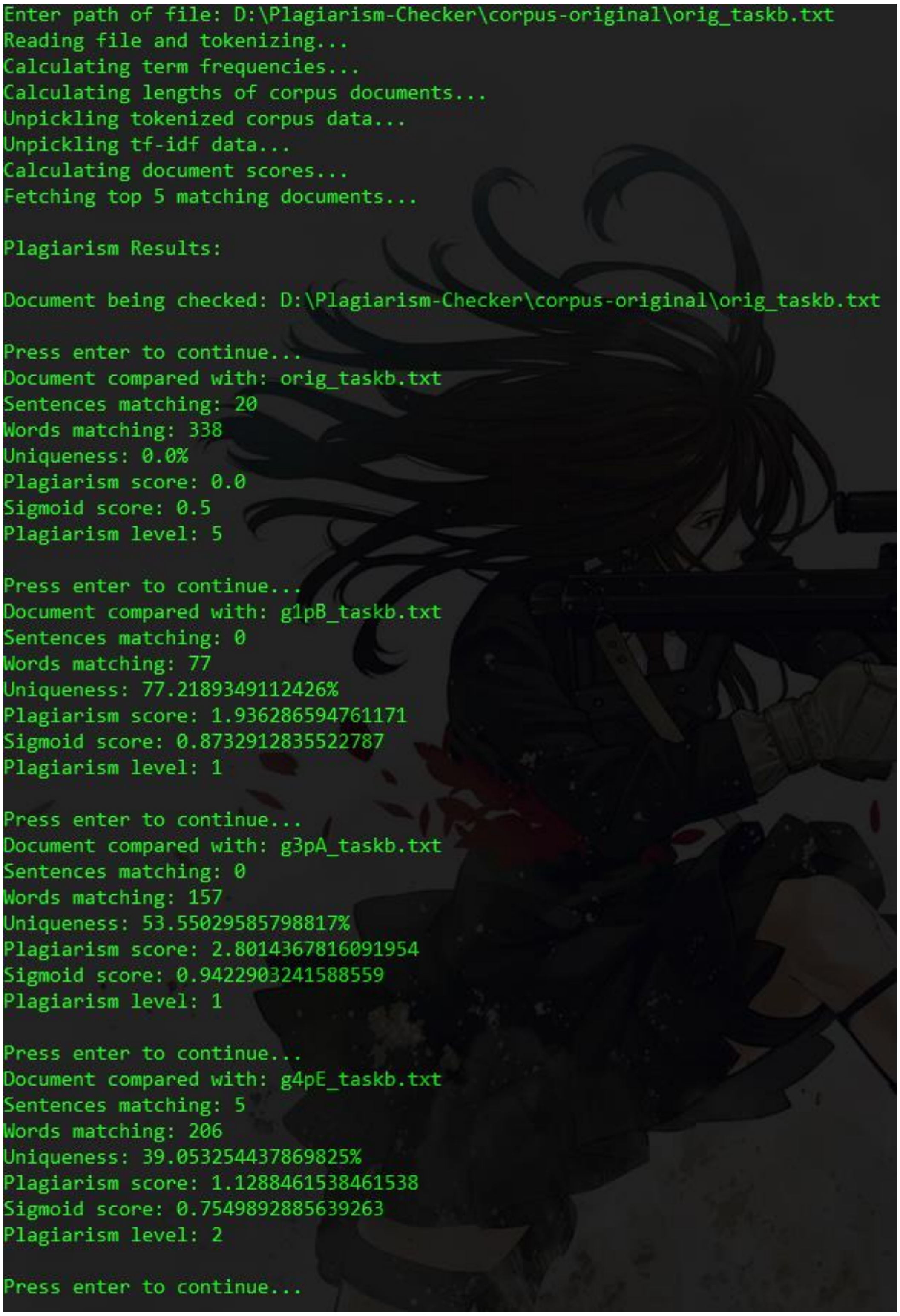
```
for each document in corpus_dictionary:
    for each term in document:
        find term_frequency
        set document_frequency:= 0
        for each other_document in corpus_dictionary:
            if term exists in other_document:
                increment document_frequency
        tf-idf = term_frequency * log(Number_of_documents / document_frequency)
        Save the tf-idf value in a dictionary with key  (term,document)
```

## Query Processing and Vector Space Model

For the query processing we used nltk to tokenize the strings and then used regex for elimination of stop words. The term frequency is then calculated form this processed query vector. The vector space utilizes the tf-idf model to calculate the cosine score, which can be simply written as the product of the tf-idf score in the document and the corresponding term frequency in the query. This is then finally used to obtain the top documents which are then in turn passed through a check8ing algorithm to ch4eck for uniqueness and plagiarism level.


Input Document
Tokenize (using nltk)
Calculate term frequency
Calculate Doc Length from corpus
Build Vector space of doc
Calculate similarity scores of each doc in corpus
Rank documents and fetch top docs
Pass top documents through checker and give comparison results

## Result



```
Enter path of file: D:\Plagiarism-Checker\corpus-original\orig_taskb.txt
Reading file and tokenizing...
Calculating term frequencies...
Calculating lengths of corpus documents...
Unpickling tokenized corpus data...
Unpickling tf-idf data...
Calculating document scores...
Fetching top 5 matching documents...

Plagiarism Results:

Document being checked: D:\Plagiarism-Checker\corpus-original\orig_taskb.txt

Press enter to continue...
Document compared with: orig_taskb.txt
Sentences matching: 20
Words matching: 338
Uniqueness: 0.0%
Plagiarism score: 0.0
Sigmoid score: 0.5
Plagiarism level: 5

Press enter to continue...
Document compared with: g1pB_taskb.txt
Sentences matching: 0
Words matching: 77
Uniqueness: 77.2189349112426%
Plagiarism score: 1.936286594761171
Sigmoid score: 0.8732912835522787
Plagiarism level: 1

Press enter to continue...
Document compared with: g3pA_taskb.txt
Sentences matching: 0
Words matching: 157
Uniqueness: 53.55029585798817%
Plagiarism score: 2.8014367816091954
Sigmoid score: 0.9422903241588559
Plagiarism level: 1

Press enter to continue...
Document compared with: g4pE_taskb.txt
Sentences matching: 5
Words matching: 206
Uniqueness: 39.053254437869825%
Plagiarism score: 1.1288461538461538
Sigmoid score: 0.7549892885639263
Plagiarism level: 2

Press enter to continue...
```

## References

1. Introduction to Information Retrieval; Christopher D. Manning, Prabhakar Raghavan, Hinrich Shutze; Cambridge

## Project Link

- https://github.com/nsurampu/Plagiarism-Checker