# ReadMe

## Prerequisites:

1. NLTk
2. Tkinter
3. Matplotlib

PS : We will be requiring pip to install the above packages

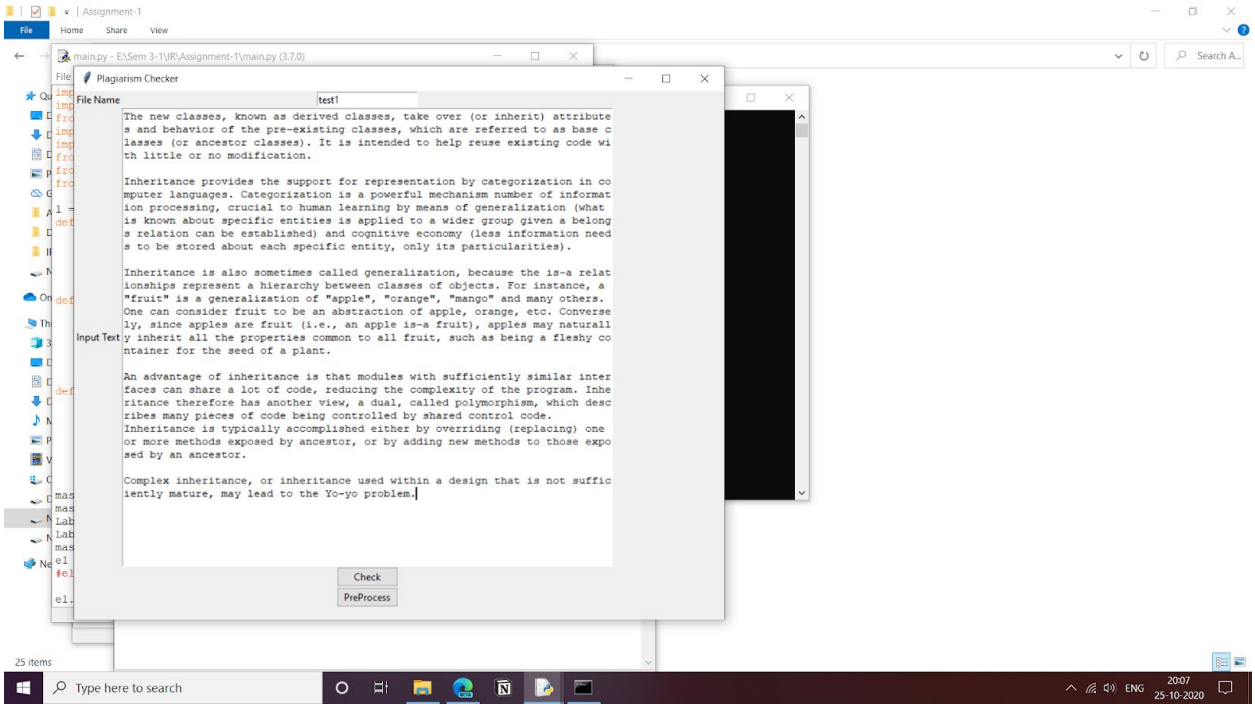
1. NLTK : pip install nltk
   We will also need to download nltk data using nltk.download()
   Ref : [Installing NLTK — NLTK 3.5 documentation]
2. Matplotlib : pip install matplotlib
   Ref : [Installation Guide — Matplotlib 3.3.2 documentation]

## Before Starting:

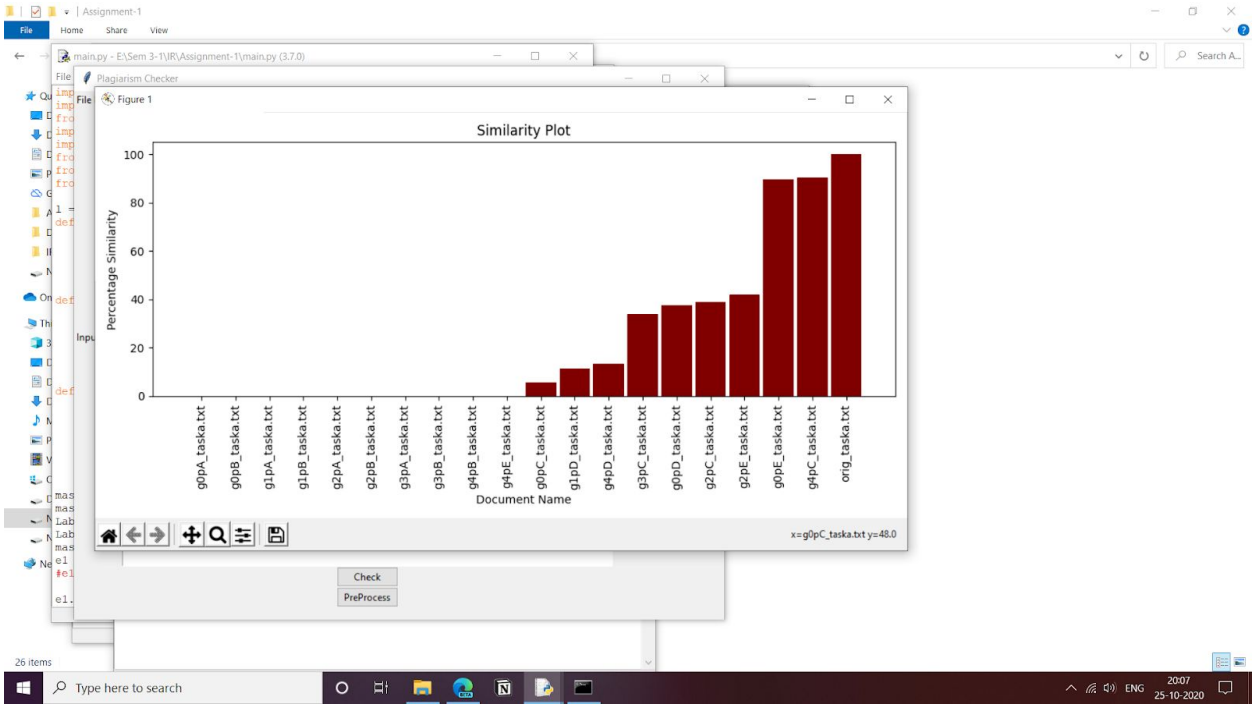
Copy the training corpus (the documents against which plagiarism will be checked) in The main.py  folder and run it.

## Here We Go:

Copy the content of the file into the text box and give the file a name as it will now be added in the training corpus. Press the preprocess button to initialize the trigrams and now we are ready to go.



Press the check button to generate a graph between the percentage plagiarism wrt to each file in the training corpus.




Now close the graph, take another file if we want to proceed without adding previous file In the corpus just click the check button.

## Limitations and Strong Points:

In case the sentences are jumbled the tri-gram methods still detects it. If the document is completely copied our model performed very well along with in the case where the text is unique. The grey areas were the places where the document has been paraphrased heavily and since the model works on a tri-gram comparison if the words are replaced with synonyms the efficiency will decrease. This can be overcomed by using a model Which detects the context of the sentences.