PLAGIARISM DETECTOR : DOCUMENTATION

PROGRAMMING LANGUAGE :

Python3

MODULES :

1. nltk : used for stemming (using Porter Stemmer), normalisation and removal of stop-words.
2. numpy : used for efficient matrix operations like addition, dot-product, multiplication
3. pickle : used to serialise the vectors in order to avoid redundant computations of useful vectors and maps.
4. wikipedia : used for the convenient scraping of Wikipedia and subsequent retreival of relevant webpages.

SOURCE-CODE :

1. preprocessing.py :

Used to perform preprocessing of the corpus documents using the 'nltk' module. Performs the following in the form of the function – **preproc**:

Parameters : document_string

i) stemming : using nltk.PorterStemmer.

ii) stopword removal : using stopwords already present in the nltk.corpus module

iii) removal of punctuation marks

iv) tokenisation : using the nltk.tokenise module

2. generate_vectors.py :

Used to generate the tf-idf vectors for each document in the corpus. The term frequencies are calculated, an inverted index is built and tf-idf scores are assigned subsequently, as described in the 'Design' section.

3. fingerprinting.py :

Use to generate the fingerprints for each document. The documents are first split into k- grams which hashed to integers using the Rabin-Karp rolling-hash function. These hash-values are further filtered using the 'winnowing' algorithm described in the 'Design' and selected to represent the true fingerprint of each document.

Two functions are defined and used:

a) winnow :

Parameters : document_string, k_value, base

Description : splits the document into k-grams and hashes the same into integer hash-values. The algorithm considers each window and uses dynamic programming to store the hash-values generated so far to calculate the hash of the surrent window.

b) get_hash :

Parameters : hash_value_array, window_size

Description : Considers windows of the hash-values of each document. In each window the least hash-value is selected along with its position in the document and in case of clash, the hash-value which is rightmost, is selected.

4. GUI.py :

Used to generate the GUI for the project using PyQt4. **The vector-space model shows the dot-product values with corpus documents and the fingerprinting-model shows the number of fingerprint matches with each of the corpus documents.**

5. scraper.py :

Used to download all the corpus documents ie, Wikipedia articles using the 'wikipedia' module.