DESIGN DOCUMENT

TOPIC: Plagiarism Checker

Group Members:

No of group members = 1

Name: Navdeep Singh

Id: 2017B5A71675H

Problem Overview:

The task is to build a plagiarism checker which will rank documents based on similarity. The program should build its indexes and IR model based on a set of training corpus and do all the pre-processing which it deems necessary. Then the model should take another document from the test set and should rank all the training set documents with respect to closeness from the test document in consideration and compute similarity with test document.

Proposed Solution:

The proposed solution is shown in the below diagram.

Therefore, the steps of solution are:

1) Extract data from dataset

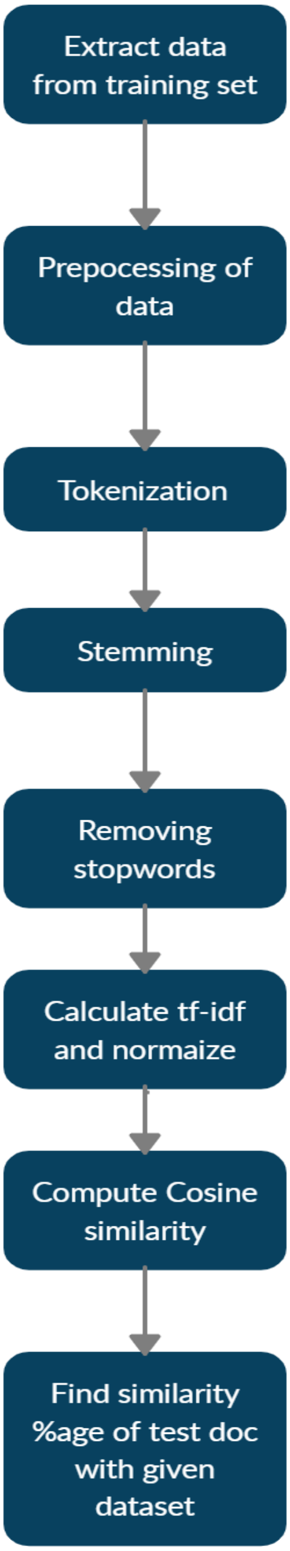
2) Pre-processing of data – tokenize, porter stemming, removing stop words(using nltk) etc.

3) Calculate term frequency, document frequency, inverse document frequency.

4) Calculate tf-idf and normalize

5) Analyze the query document and calculate cosine similarity between query and training documents.

6) Arrange training files in decreasing order of similarity w.r.t test document.


Extract data from training set
Prepocessing of data
Tokenization
Stemming
Removing stopwords
Calculate tf-idf and normaize
Compute Cosine similarity
Find similarity %age of test doc with given dataset

Data structures used:

Mainly vectors and matrices are used for tf ,df, idf, tf-idf arithmetic and storing of results

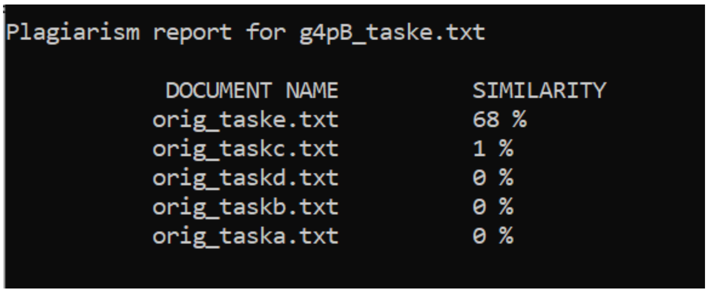
Arrays have been used for storing names and contents of files, similarity percentages and other final results.

Results on sample dataset:

Dataset from the website http://ir.shef.ac.uk/cloughie/resources/plagiarism_corpus.html was used as corpus for testing my plagiarism checker.

All 5 files of the corpus were used for training set. For query, 'g4pB_taske.txt' file was used.

The results are as follows:



```
Plagiarism report for g4pB_taske.txt

            DOCUMENT NAME           SIMILARITY
           orig_taske.txt           68 %
           orig_taskc.txt           1 %
           orig_taskd.txt           0 %
           orig_taskb.txt           0 %
           orig_taska.txt           0 %
```

The result gives 68% similarity with 'orig_taske.txt'. This result matches well with the result given on website which states that there is very high amount of plagiarism in this test file w.r.t 'orig_taske.txt'.

Run time of the program: ~ 3 seconds

Limitations

1) The main limitation is that word order which is an important part of understanding the meaning of a sentence is not considered.

2) If someone changes most of the words of the sentence with synonyms keeping the meaning same then plagiarism checker won’t be able to detect plagiarism.

3) Also, document length can introduce a lot of variance in the TF-IDF values and thus final results.

4) If the corpus is large, then the program takes a large amount of time to execute.

5) Plagiarism detector won’t be effective if document contains large amount of stop words.