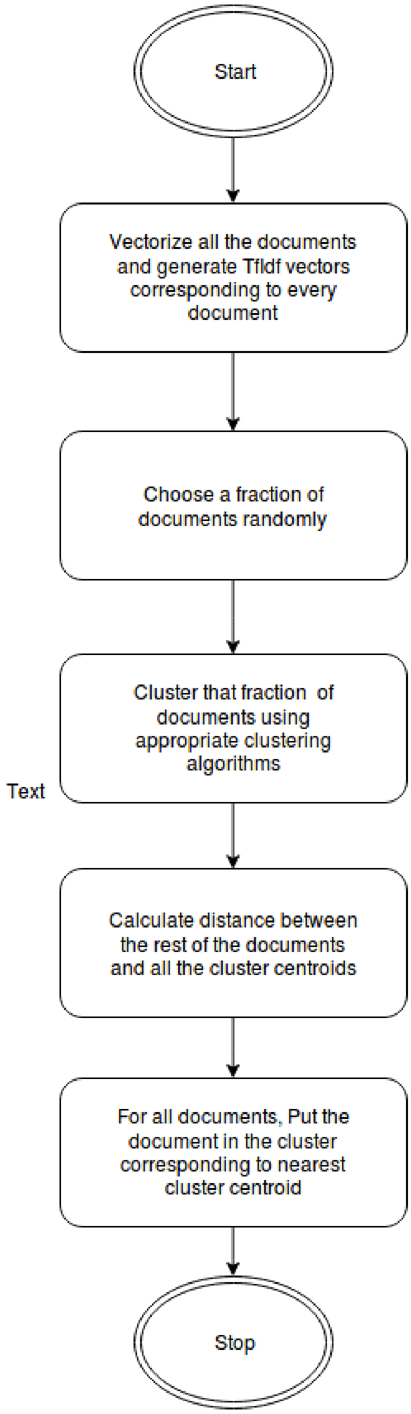
Large-scale clustering of documents


Start
Vectorize all the documents and generate TfIdf vectors corresponding to every document
Choose a fraction of documents randomly
Cluster that fraction of documents using appropriate clustering algorithms
Text
Calculate distance between the rest of the documents and all the cluster centroids
For all documents, Put the document in the cluster corresponding to nearest cluster centroid
Stop

Basic Concept:

First, we will have to generate vectors from given text documents. Here we chose **TfIdf** vectorizer.

Then we chose a fraction of documents **randomly** and cluster them using suitable algorithms. Here we chose Kmeans.

Data Structures and Algorithms used:

Here we have used following data structures in the code: 1D arrays, 2D arrays, Dictionaries.

The Algorithm used is described in the adjoining flow diagram.

Why we used **TFIDF**? Because TFIDF not only emphasises on the words that are sufficiently abundant but also ignores the ones which are overly frequent. For example, if 90% documents have the word ‘that’, TFIDF will not consider that as an important word for clustering because that would favour misprediction.

Why we choose documents **randomly**? Because the probability that we have chosen some documents that belong to each cluster is high if chosen randomly. That yields better performance.

Dataset and parameters:

We have tested on BBC’s dataset. Although, we are sure that the classifier will work equally well on other unbiased datasets.

Input Format:

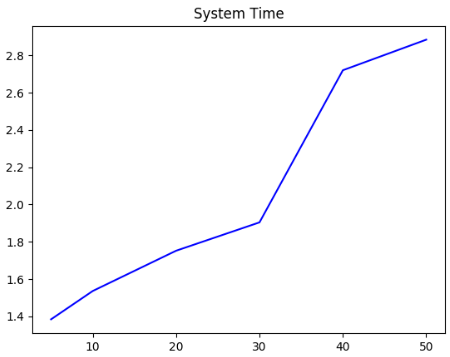
<path to folder containing the folders named as ‘labels’ and each folder containing the text files which belong to corresponding labels>
<path to folder to dump output>

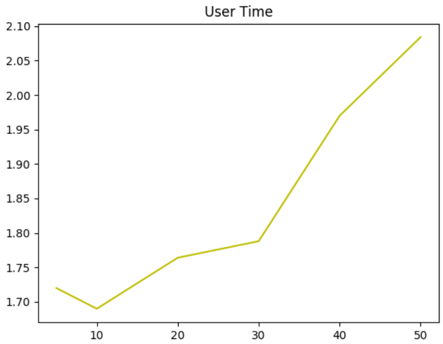
Output:

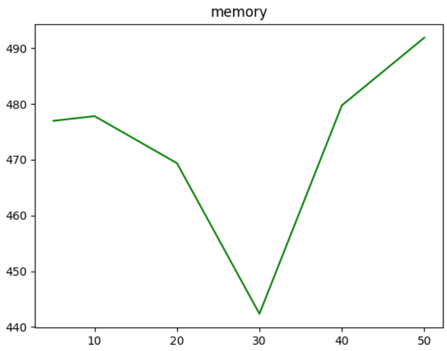
‘cluster.txt’ contains clustering map of files
‘resourses.txt’ contains log of used resources

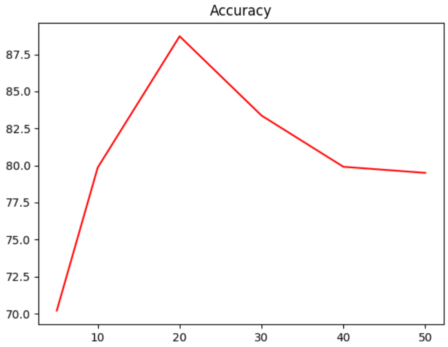
Parameters: The program can be tuned using following parameters:
Factor of documents taken for generating clusters, Parameters in TFIDF vectorizer.

Results:









In general,

- amount of memory used increases as the fraction used for clustering increases.
- time required, both user and system increases as the fraction used for clustering increases.
- accuracy however shows an optima at 20%.

Contribution:

Ashutosh : idea, implementation, finding accuracy, plot graphs, report.
Amish: idea, implementation, finding resources.
Vishal: idea, implementation, finding accuracy, plot graphs

Each member contributed in his own unique way. Some took initiative, some gave ideas, some found flaws in ideas: helping to improve the idea, some worked late till 5 am, some worked after getting up at 5 am. Everyone was great, project was fun.

Report made by Ashutosh