Team:

•Kshitij Grovor 2015A7PS070H

•Bhavathi Reddy 2015A7PS020H

•Meghana Kumar 2014B5A7932H

Problem Statement : To implement a retrieval system based on vector space model on the given dataset

Language : Python v3.6.1

Working :

1. Documents are processed in the Data_Work.py file and Tokenisation and Stemming are performed.

2.Then a data structure of a list of dictionaries with a list as the value of the key:value pair is created. Each element of this list has the number of occurances of that term.

3.Heuristics based sentiment analysis is performed on every document and degree of positivity/ negativity is stored during preprocessing

4.Term frequency of each term in each document is first normalized and then the tf idf (term frequency - inverse document frequency) score is calculated

5.The idf of every word is calculated using log(N/df) where N is the size of the corpus and df is the document frequency of the word. The tf of the word per document is calculated by the formula tf/(sqrt(sum(tf(i)^2))) where tf is the frequency of the word in a particular document. The formula used for weighing the document-query similarity is nnc.ntc (ddd.qqq). The document vector (which has only the tf) is normalised by making it as a unit vector at runtime.

6.On being given a new query, the tfid score of the terms of the query is calculated and the cosine similarity is found and ranked

7.Tkinter GUI of python is used for making it intuitive to use.

8.The user has the option to order the relevant queries by positive or negative sentiment.

Setting up:

1.Extract the folder and run GUI.py. Make sure you have sklearn and nltk libraries installed correctly before running the project.

2.It takes around 6 minutes to pre-process the data.

3.Running time is usually of the order 0.01 seconds

Screenshots