Main Module

This module is responsible for preprocessing which includes tokenization, normalization of the corpus.

We declare dictionaries for each separate emotion. 7 in total.

First the module picks up the ISEAR data-set which is present as a csv file. Then it takes in the string, and its label – one at a time. Parallelly we keep a hash-set of all the distinct words we have encountered till now. If a new word is encountered, we initialize the entry in all the dictionaries with respective frequencies. If the word is present in the hash-set, we simply update the frequencies.

Then we import the TF IDF module and calculate the IDF for each word by passing the list of dictionaries.

Then we pass each document dictionary in a loop and calculate the TF-IDF for each word wrt to each document and put it in a TF-IDF 2D matrix.

Subsequently we store the distinct words list into ‘distinct\_words.pkl’ file and TF-IDF matrix into ‘tf\_idf.pkl’ for other files to retrieve them.

TF IDF Module

For computing tf-idf, we used 3 functions.

The first function named "computeTF" computes the term frequency score for each word in the corpus. It divides the count of that particular word in the document with the total number of words in that document. It takes the word dictionary as the argument and returns the dictionary of words with their TF score for each word for each document.

The second function named "computeIDF" computes the Inverse Data Frequency for each word. It takes the list of documents as argument and returns the IDF score for each word.

The third function named "computeTFIDF" computes the TF-IDF score for each word, by multiplying the TF and IDF scores. It takes the returned components of first two function as arguments and returns the matrix having TF-IDF score for each word with corresponding to each document.

Tweet Scraping Module

This module works on a set of *xlsx* files present in the current directory in a certain format found in the dataset and extracts the tweets. It then processes them for sentiment analysis by removing the stopwords, cleaning and tokenizing the lexioms.

**Input format:** One *xlsx* file each for a NASDAQ 100 company containing tweets and a pickle file ‘distinct\_words.pkl’.

**Output format:** The output is a set of pickle files one for each company, containing the tokenized tweets.

Interactive Module

This module accepts a *String* from the user and processes it by removing special characters, tokenizing it and uses *NLTKs SnowballStemmer* tostem the tokens.

It imports the distinct words list from ‘distinct\_words.pkl’ file and converts the tokenized list into a proper vector with the length equal to the number of distinct words. For each distinct word, a 0 or 1 is placed in the vector depending on whether the word is present in the query or not.

Then it imports the TF\_IDF vector from ‘tf\_idf.pkl’ file.

Then it does the matrix multiplication of the vector with the TF\_IDF vector to get a final 1\*7 matrix which contains the final scores with respect to each emotion.

Then it does a sorting on the basis of scores and gives out the first 3 results.

In case, no result could be determined, it gives out a message, with the same.

Visualization Module

In this module, we test a use-case for our sentiment analysis, i.e. finding a co-relation between sentiments and stock prices.

We load the distinct words, TF IDF matrix and the tweets for any company – inn our case – American Airlines.

We process the tweets one by one similarly as the query processing in the Interactive Module, and finally make the vector w.r.t. the distinct words. We multiply it with the TF-IDF vector to get the 1\*7 answer matrix. Now we add those values to the row corresponding to the date in the ‘data’ list.

After filling in the ‘data’ list, we find the ratio of each emotion on any given date.

We also import the stock prices for American Airlines for the same date-range. We fill in the missing values for the weekends with the last available value.

Finally we plot the stock prices variation and the sentiment variation to eye-ball any co-relation.