Navigating Project

Python Code in Red

Unformatted Text

Formatted/Cleaned Text

Word Cloud

# Data Folder

1. getQueryTweetsRawFromTwitter.py

* This python code gathers data from Twitter with the Stream API using specific keywords. It saves just the Tweets to a file. It appends each time it retrieves a tweet to the file, labelled ‘queryTweetsRaw.txt’.

1. queryTweetsRaw.txt

* The result of running ‘getQueryTweetsRawFromTwitter.py’. The data is unformatted and not cleaned yet. Duplicates are also very much prevalent in this data file. 8689 Tweets.

1. cleanQueryTweetsFormatted.py
2. Takes the ‘queryTweetsRaw.txt’ file and applies extensive cleaning to it. It removes duplicates. Fixes asci/Unicode issues. Removes punctuations. Gets rid of Usernames and website links. Removes Hash tags (But not the word after the hash symbol.) Also some apostrophes are taken care of using common usage of certain words in the English Language. It outputs ‘queryTweetsCleaned.txt’.
3. queryTweetsCleaned.txt

* The output from ‘cleanQueryTweetsFormatted.py’ is the cleaned up, no duplicated, formatted text for processing. 6488 Tweets reduced from 8689.

# Classification

1. trainedData.txt

* This file is the training data that is formatted as follows: [id Number, True/False (1/0), Text]. This file is 3000 tweets that are used to train the LR Algorithm.

B) queryTweetsLR.py

* The LR python code that takes in ‘trainedData.txt’ as the training set. It uses a list of words, referred to as ‘stopwords’ and is a list of words that are not relevant to the tweets we are after. The word frequency is 50 due to the large number of tweets and trying to avoid common words that sneak past the stopwords list. With this the LR is built and we feed it in ‘queryTweetsCleaned.txt’ and it returned 176 positive Tweets in the following format: [Id, Text]. It saves these tweets to the file ‘positiveQueryTweetsLR.txt’. It also creates a file to be used for creating word clouds, ‘positiveQueryTweetsLRForWordCloud.txt’. This file is unformatted because of the nature of the word cloud maker; we do not need the text to be formatted.

C) positiveQueryTweetsLR.txt

* These tweets are positive from the ‘queryTweetsLR.py’. They are in the format [Id, Text]. These will eventually be used in the creation of clusters of data.

D) positiveQueryTweetsLRForWordCloud.txt

* This file is created from the ‘queryTweetsLR.py’ code. It is unformatted because it is going to be fed into a website to give us a word cloud of the most used words.

E) wordcloud.bmp



From the word cloud we can identify certain words that stand out over other words. The biggest two are ‘Sneezing’ and ‘flu’. This follows in our group project idea of people who tend to tweet that they are sneezing do not associate that with the flu versus those who tweet about the flu not tweeting related to sneezing. This is more apparent in the word clouds of the clusters. Because we are focusing on two specific topics the clustering was done to two groups to reflect this.

# Clustering

1. clusterPositiveLRTweets.py

* Takes in ‘positiveQueryTweetsLR.txt’ as the data to cluster. Using two clusters and setting word frequency to 30 or more. Also using the same ‘stopwords’ as used in the ‘queryTweetsLR.py’. This produces two very nice clusters. ‘Cluster-0’ is of 115 tweets, relating to the Flu. While ‘Cluster-1’ has 60 tweets relating to sneezing. This program also produces two related files for use in a word cloud maker ‘cluster-0WordCloud.txt’ and ‘cluster-1WordCloud.txt’. Both sets are unformatted.

B) cluster-0.txt

* This file contains 115 tweets clustered together based on ‘flu’.

C) cluster-1.txt

* This file contains 60 tweets clustered together based on ‘sneezing’.

D) cluster-0WordCloud.txt

* Unformatted text of ‘cluster-0.txt’.

D) cluster-1WordCloud.txt

* Unformatted text of ‘cluster-1.txt’.

E) wordcloudFlu.bmp



If you look we see flu and cold as the biggest two words. This will hold true when it comes to the association part of the project.

F) wordcloudSneezing.bmp



Sneezing and day are the two biggest here and will also hold true for association.

# Association

A) associationTokenCluster0.py

* Takes in the cluster text file and filters out all the words that are not in the top word list. There are two versions of this file, one for each cluster text file. The format is to take each sentence and just keep the keywords, each in quotations marks and separate each file by a comma so it can be used in the association program. This python code produces ‘cluster0PositiveLRAssociateFormat.txt’.

B) associationTokenCluster1.py

* Takes in the cluster text file and filters out all the words that are not in the top word list. There are two versions of this file, one for each cluster text file. The format is to take each sentence and just keep the keywords, each in quotations marks and separate each file by a comma so it can be used in the association program. This python code produces ‘cluster1PositiveLRAssociateFormat.txt’.

C) cluster0PositiveLRAssociationFormat.txt

* A list of the keywords from the positive LR tweets. Used in associating words in the ‘cluster0Association.py’ program. Need to copy this by hand into the program and remove the last comma to avoid errors. Related to flu.

D) cluster1PositiveLRAssociationFormat.txt

* A list of the keywords from the positive LR tweets. Used in associating words in the ‘cluster1Association.py’ program. Need to copy this by hand into the program and remove the last comma to avoid errors. Related to sneezing.

E) cluster0Association.py

* Associates words in the cluster0 list. Using .3 minimum support we are able to find association between flu and cold. This is copied by hand to the ‘cluster0AssociationRule.txt’ file by hand. It lists each tweet as a user and if there is a recommendation for association.

F) cluster1Association.py

* Associates words in the cluster1 list. Using .3 minimum support we are able to find association between flu and cold. This is copied by hand to the ‘cluster1AssociationRule.txt’ file by hand. It lists each tweet as a user and if there is a recommendation for association.

G) cluster0AssociationRule.txt

* At the top is the association with a .3 support and lists the confidence. It recommended for 52 users either sneezing->day or day->sneezing.

H) cluster1AssociationRlue.txt

* At the top is the association with a .3 support and lists the confidence. It recommended for 116 users either flu->cold or cold->flu.