CS410 Project Documentation

Twitter Sentiment Analysis

Rudy Rath

Overview

The objective of this project is to perform Sentiment Analysis of Tweets for a particular topic and the steps involved are –

1. Collect Twitter data
2. Process the data
3. Perform sentiment analysis

Two approaches are used –

Approach A

This approach extracts Twitter data in real time by taking user input and then Sentiment Analysis is done

Approach B

This approach uses previously extracted Twitter data for Sentiment Analysis

Implementation and Usage

The environment used is **Jupyter** notebooks running on an **Anaconda** installation.

Approach A

This approach involves extracting Twitter data in real time by taking user input and then performing Sentiment Analysis.

The following tools are used for Approach A

- **Tweepy** package for extracting Twitter Data
- **NLTK**
- **VADER (Valence Aware Dictionary and sEntiment Reasoner)**
- **TextBlob**

Prerequisites

To extract Twitter Data in real time there are prerequisite steps that have been to be completed –

Creation of Twitter Developer Account

Instructions for this are available at https://developer.twitter.com/en/support/twitter-api/developer-account

Generation of Twitter api key and access token for application

This step requires –

1. **Creation of a Project within the Twitter Developer Account** - This can be done by using the information provided at https://developer.twitter.com/en/docs/projects/overview

2. **Creation of an Application in the project** - This can be done by using the information provided at https://developer.twitter.com/en/docs/apps/overview

3. **Generation of api key and access token** - This can be done by using the information provided at https://developer.twitter.com/en/docs/authentication/oauth-1-0a/api-key-and-secret

Implementation

The code is implemented as follows

1. Twitter api key and access token are stored in a config file
2. **Configparser** is used to read the Twitter api key and access token from the config file
3. Twitter api key and access token are then loaded onto the Tweepy package
4. User is prompted for a keyword to search
5. Twitter data is scraped for that keyword using **Tweepy** package and stored in a data frame and also a csv file for future use
6. **SentimentIntensityAnalyze**r from **VADER** is used for sentiment analysis
7. Subjectivity and Polarity of tweet text are calculated and scores are assigned to tweets
8. Based on scores, tweets are categorized to Positive, Negative and Neutral
9. Count of Positive, Negative and Neutral tweets are displayed
10. Bar graph of the count of Positive, Negative and Neutral tweets
11. Word cloud of all tweets is displayed using stop words

Usage

Software code files are

- **CS410Project-TwitterSentimentAnalysis-ApproachA.ipynb -** *This is the main notebook for execution*
- **twitter-api-sample.cfg –** This is a sample configuration file that is parsed to get the Twitter api keys and access token

To execute the code

1. Download the above files to local directory
2. Edit **twitter-api-template.cfg** and enter your api key, api key secret, access token, and access token secret in the placeholders provided. Save the file as **twitter-api.cfg**
3. Run the Jupyter notebook **"CS410Project-TwitterSentimentAnalysis-ApproachA.ipynb"** after making sure the **twitter-api.cfg file** is in the same directory
4. Input the keyword for the search phrase when prompted during runtime
5. Review results

Approach B

This approach involves using an existing extracted dataset of tweets user input to perform Sentiment Analysis.

The following tools are used for Approach B

- **NLTK**
- **VADER (Valence Aware Dictionary and sEntiment Reasoner)**
- **TextBlob**

To illustrate this approach, I'm using the data set that was extracted in real time using Approach A above (The phrase for this extraction was "**Biden**") and stored in the "**extracted_tweets.csv**" file

Implementation

The code is implemented as follows

1. The csv file is loaded into a data frame
2. **SentimentIntensityAnalyze**r from **VADER** is used for sentiment analysis
3. Subjectivity and Polarity of tweet text are calculated and scores are assigned to tweets
4. Based on scores, tweets are categorized to Positive, Negative and Neutral
5. Count of Positive, Negative and Neutral tweets are displayed
6. Bar graph of the count of Positive, Negative and Neutral tweets
7. Word cloud of all tweets is displayed using stop words.

Usage

Software code files are

- **CS410Project-TwitterSentimentAnalysis-ApproachB.ipynb -** *This is the main notebook for execution*
- **extracted_tweets.csv –** This is the csv data set that was extracted in real time using Approach A above (The phrase for this extraction was "**Biden**")

To execute the code

1. Download the above files to local directory
2. Run the Jupyter notebook **"CS410Project-TwitterSentimentAnalysis-ApproachB.ipynb"** after making sure the **extracted_tweets.csv** file is in the same directory
3. Review results

Note on Implementations

My original idea was Approach A where the tweets are extracted based on user input and Sentiment Analysis is performed on that.

Since this approach requires a Twitter Developer account and the Twitter api key and access token are linked to my developer account and are not supposed to be shared publicly, I had listed this as one of the challenges in my Progress Report.

One of the reviewers advised that I could use publicly available data sets.

TA's advice was to write an alternate demo code only for the reviewers that takes the data from a saved csv/txt file and runs the later part of my code on that.

Considering the above inputs, I'm providing both approaches A and B so that the reviewers have options on how they want to run the code.

- Approach B is quicker as the csv file is available and one can just run the Jupyter notebook to confirm code is being executed correctly.

- Approach A will take longer way as it requires a Twitter Developer account and Twitter api key and access token for an application of that Developer account which need to be entered into the config file.

Project Deliverables

GitHub Repo

https://github.com/rudyrath/CourseProject

Software Code

Software code files for Approach A and Approach B are listed in the above sections under "Usage" header.

All files are available in the GitHub repo

Software Documentation

This File : **CS410 Project Documentation_RudyR.pdf**

Available in the GitHub repo

Software Presentation

Presentation slides (available in the GitHub repo): **CS 410 Project Presentation_RudyR.pptx**

Presentation Video link : https://mediaspace.illinois.edu/media/t/1_eizzv6nv

NOTE: Presentation Video Link also added to Readme of GitHub Repo