TWITTER SENTIMENT ANALYZER

[CIS 496H: NATURAL LANGUAGE PROCESSING]
[TERM PROJECT]

Submitted by: Radha Satam [300130632]

Instructor: Gabriel Murray

Table of Contents

Introduction

The gamification project titled **"Twitter Sentiment Analyzer"** aims to provide a fun way to collect gold-standard sentiment labels from users by showing them tweets containing specific queries. It also does sentiment analysis for 500 most recent tweets about any query specified by the user.

Working

System Perspective

The system is trained using Naïve Bayes Classifier on some gold standard training data. The training dataset (train.csv) consists of tweets already tagged as – "positive", "negative" and "neutral".

Once learned, the system then prompts the user to enter a query search term. This query is then used to find the 500 most recent tweets about the search term, and stored in a csv file (tweet.csv). After this, the sentiment is analyzed for all 500 tweets using the training dataset as the gold standard labels. This sentiment analysis of all tweets is displayed to the user – positive, negative or neutral.

The game display then picks a tweet from the queried tweets, displays it to the user and then asks the user to tag the tweet as positive, negative or neutral. After this, the result is displayed (User's response and System's response) and displays if the tag given by the user matches the tag analyzed by the system or not.

If the responses match, then this tweet is added to the training dataset (train.csv). If it does not match, but the general sentiment of the 500 tweets that are analyzed does match, the tweet and the sentiment is added to the training data. For example, if the analysis of the system for the 500 tweets is positive and for the tweet that is shown is negative, but the user responded positive. Then the tweet also gets added as a gold-standard label with the user's response, assuming that the system tagged data was wrong. (The user isn't notified of this though)

User Perspective

The user will be prompted to enter a query search term. For example, any brand name or event.

The sentiment of that query from the last 500 tweets is then analyzed and displayed to the user. (Positive, negative or neutral)

A tweet is then displayed from the 500 tweets that are extracted.

The user is asked to tag the tweet. "P" for positive, "N" for negative and "X" for neutral.

The result is displayed to show if the user's response matched the system response for that tweet or not.

The user can press "Y" to display another tweet from the same query or "E" to enter the query again.

Instructions for running the game

I preferred making a video for showing the working of this project instead of attaching screenshots and thought it best to provide instructions in the same video. Here's the link -

https://youtu.be/WZn3rKSQouI

Project Files with descriptions

README.md

Provides a short description of the project along with the instructions for use

GetTweets.py

Makes use of the **tweepy** package. It includes a function that takes the query as the argument, connects to Twitter API and retrieves the 500 recent public tweets using the query provided as input and stores it in a csv file. (tweet.csv)

SentimentAnalyze.py

Makes use of **nltk** package. It includes several functions in order to extract the data from the training dataset (train.csv) and then process that data. Naïve Bayes Classifier is used to classify the data. Laurent Luce's blog had great information about this. *See reference below

It's also used to classify the test data contained in (tweet.csv) after the user has entered a query. It returns the overall sentiments of the 500 tweets, the list of 500 tweets and also the list containing the sentiment labels for the 500 tweets.

tokenize.py

The functions in SentimentAnalyze.py make use of this file in order to tokenize the tweet strings. It preserves the emoticon strings and the html strings which may be in the tweet. *See reference for the tokenize class below

Main.py (pygameDemo.py)

This is the main file which contains the code for the game. The user interface is generated here. The actions are specified based on certain events and based on which screen the user is currently on. This file makes use of the pygame package and uses GetTweets to get the tweets based on user query and SentimentAnalyze for training and tagging the dataset.

train.csv

The training dataset contains 600 positive and 600 negative tweets. It's tagged like -

0 – Negative 2 – Neutral 4 – Positive *See reference to the training dataset below

tweet.csv

This file is used to contain the 500 tweets from the query entered by the user. GetTweets.py has a function that does that.

Packages required

Nltk, Tweepy, Pygame (Click to go to the downloads page)

Steps used in development

I used the following steps to develop the project. They helped a lot in keeping the process organized.

STEP 1 – Figure out how Sentiment Analysis is implemented in Python – find a training dataset

STEP 2 – Figure out how to get Twitter data on Python using the API

STEEP 3 – Get it to display n number of sample tweets based on any keyword entered by user.

STEP 4 – Analyze tweet sentiments using gold standard labels of words

STEP 5 – Allow users to input the sentiment and compare it to the result from the system

STEP 6 – Gamify it

Issues & scope for improvement

[.exe] for the game – I used cxFreeze package but couldn't get it to create an .exe for the game.

Improving the game design – user interface as well as the functionality (having points or scores, having more users and comparing their results). I haven't developed a game before so it was a bit of a learning curve to wrap my head around the fact that it runs as frames.

Optimizing the code for reducing processing time – It takes quite a bit of time for training the system every time a query is entered.

Including **more training data** for better results; currently it's not always accurate.

Neutral data – I couldn't get any neutral data since the dataset I found had over 15 million tuples in a csv file and was too large to open.

I encountered extraction error from Tweepy a couple of times. It does so if I try to extract a lot of tweets, it prints an error that the limit is reached. It works fine again after waiting for a couple of minutes.

Links

Code on GitHub – https://github.com/RadhaSatam/Gamification---Twitter-Sentiment-Analyzer/

Instructional video – https://youtu.be/WZn3rKSQouI

References

Mining Twitter Data

Bonzanini, M. (2015, March 2). *Mining Twitter Data with Python.* Retrieved from http://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/

Pygame

Pygame Events Reference. (n.d.). Retrieved from Pygame.org: https://www.pygame.org/docs/ref/event.html

Harrison. (2014, November 10). *Pygame (Python Game Development) Tutorial Series.* Retrieved from YouTube Channel - thenewboston: https://www.youtube.com/watch?v=K5F-aGDIYaM&list=PL6gx4Cwl9DGAjkwJocj7vlc_mFU-4wXJq

Sentiment Analysis

Alba, J. (2012, November 1). *Basic Sentiment Analysis with Python.* Retrieved from fjavieralba: http://fjavieralba.com/basic-sentiment-analysis-with-python.html

Luce, L. (2012, January 2). *Twitter sentiment analysis using Python and NLTK.* Retrieved from http://www.laurentluce.com/posts/twitter-sentiment-analysis-using-python-and-nltk/

Tokenize

Potts, C. (2011). *Twitter-aware tokenizer.* Retrieved from http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py

Twitter Dataset

Link to download corpus. (n.d.). Retrieved from Stanford Edu: http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip

Sentiment140. (n.d.). *Tweet Corpus for Academics*. Retrieved from http://help.sentiment140.com/for-students

Other references – Stackoverflow (extensively), **Pygame website,** other **YouTube** videos, **NLTK book**