Julián Niebieskikwiat Godoy
A01207868
23 / Nov / 2016

Challenge Report: Sentiment Analysis of Twitter Data

Justification:

As we know, times have changed, nowadays people express their feelings, emotions and opinions about a particular topic in social networks. Twitter right now is not as popular as it was a few years ago, but it's still a good source of information to know what people is thinking.

This project has the main goal to classify a tweet (or any text) as "positive" or "negative" in terms of a sentiment. Then with this data we can see the impact that a topic is having on social networks. It will be hard for a human to read all the tweets the users are writing and classify them in real time, so that's why the best option to achieve this is using the computational power we have right now. An AI was trained with a labeled dataset to help us in this task, also a connection with twitter was implemented to get information in real time and plot it to make it easier to understand.

The project was divided in 6 main modules:

- Labeled Dataset: The dataset for this project can be found in this source: https://github.com/caesar0301/awesome-public-datasets it contains 1,600,000 tweets with the emojis removed and labeled as "positive" or "negative".
- Clean Data: The chosen dataset contained a lot of features, some of them were not giving enough information to this problem, as "username", "date", "time", and so on. A script called "clean_data.py" was made to automatically clean the dataset features and generate two more files called "cleaned_training_data.csv" and "test_data.csv" to train and test the algorithm.
- Training algorithm: A Naive Bayes Classifier was chosen for this project for training. Naive Bayes works with classes, each class is described by a vector of features. In this case, classes are "positive" and "negative" and their features are the significant words of the labeled tweets. Each word has a specific weight depending on their frequency in the texts.
- Predict: After training the classifier, this script loads the trained data and let us predict the classification of any text without having to train the classifier each time.
- Accuracy: A script was created for testing the accuracy and with 10,000 tweets for training and 1,000 tweets for

testing there was an accuracy of 74%. This might be improved by increasing the training data and using better methods to clean the tweets like Porter Stemming.
- Twitter Connection: A script that opens a connection with twitter to be listening to incoming tweets with a specific hashtag or keyword. This will help anybody to track their impact in this social network.

It's worth to mention that the system will have errors classifying "tricky" tweets like: "I love being sad". But also we as humans are not able to classify it because it's a paradox.

As it was mentioned before, the system accuracy could be improved by adding better algorithms to clean the tweets like Porter Stemming which converts a word to its root and helps to subtract better features for each class.

The system can scale for to more than just two classes, and instead of classifying only "positive" and "negative" there could be multiple categories like "happy", "bored", "depressed", "excited" and so on. There is a similar online application that does this and it's called Sentiment Viz.

Documentation:

The project source code can be found in the following Github repository:
https://github.com/juliannieb/IAFinalProject

To download it just install Git in your computer and type:
git clone https://github.com/juliannieb/IAFinalProject.git

It's very simple to run the program. All scripts are written in python 2.7 and to avoid compatibility problems, a file called "requirements.txt" was created. It's recommended to create a virtual environment where all the requirements of the project can be installed without modifying the global python environment. Instructions to install and create the virtualenv can be found in the official site https://virtualenv.pypa.io/en/stable/.

After installing, creating and activating the Virtual Environment you must be able to install all the requirements by running the following command in the terminal inside the project folder:
pip install -r requirements.txt

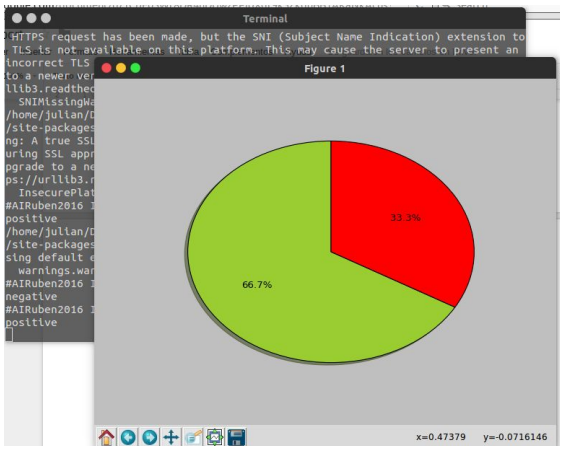
The official dataset for this project was not added to the repository, if you want to run the clean data script you must download it and add it to the project folder. It can be found in the following link:
http://help.sentiment140.com/for-students/

After doing all previous steps you might be able to start running the python scripts. Running examples are:

python clean_data.py
python train_sentiment_tw_analysis.py
python predict.py
python test_accuracy.py
python twitter_listener.py

The clean_data.py script has variables to manage the amount of positive or negative tweets you want for training and testing the classifier. You can modify those values for your convenience.

The Twitter listener script will open automatically a connection with the Twitter API and start listening to the provided hashtag. There is a variable to manage the hashtag that you want to follow. It can be modified to any topic you want to follow. After running this script, another window will be opened to plot the results in real time, so you will be able to interpret it easier.



References:

Sentiment 140. (n.d.). Retrieved November 14, 2016, from http://help.sentiment140.com/for-students/

Awesome Public Datasets. (n.d.). Retrieved November 14, 2016, from https://github.com/caesar0301/awesome-public-datasets

Murphy, K. P. (2006, October 24). Naive Bayes classifiers. *Naive Bayes Classifiers*. Retrieved November 17, 2016, from https://datajobs.com/data-science-repo/Naive-Bayes-[Kevin-Murphy].pdf

Porter, M. (n.d.). The Porter Stemming Algorithm. Retrieved November 20, 2016, from https://tartarus.org/martin/PorterStemmer/

Bonzanini, M. (n.d.). Mining Twitter Data with Python (Part 1: Collecting data). Retrieved November 21, 2016, from https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/

Matplotlib. (n.d.). Pyplot Documentation. Retrieved November 21, 2016, from http://matplotlib.org/api/pyplot_api.html

Sentiment Viz. (n.d.). Retrieved November 21, 2016, from https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/