Analysis of Tweet Sentiment with Natural Language Processing (NLP) and Naive Bayes

Machine learning models allow us to process vast amounts of data quickly and build models which attempt to understand the data. For example, email systems use NLP as part of their spam detection systems. Many social networking platforms such as Facebook and Twitter use such systems to flag offensive posts for review or elimination from the site. Success of such systems is paramount to elimination of undesirable content while allowing desired content to flow unimpeded.

In this project you will attempt to analyze a collection of tweets to determine whether the messages carry a positive or negative sentiment using a Naive Bayes classifier. Given a collection of positive and negative labeled tweets your model must process and analyze the text and learn how to classify a given tweet as having a positive or negative sentiment. Then, you will have a trained system that you can use to analyze some tweets around the election to find trends.

Hints and Tricks:

Tasks you ***may need to*** perform:

- Determine the features of the message which are used for sentiment analysis i.e. words or frequency of words which matter.
- Process raw tweets messages and create a dictionary of words which appear in tweets.
- Make encoded tweet features i.e. fixed length vectors for all tweets.
- Train model from encoded features and sentiments.
- Predict outcome for new unseen tweets.

Example of feature selection:

- Clean up text (Remove retweet tags, punctuations, …)
- Remove all words that are likely to be irrelevant (a, an, the, ...)
- Stem words (Find common root for words i.e. generally bird and birds should be counted as bird in the dictionary)
- Count the association of all words with positive and negative sentiment
- Encode the tweets. For example tweet features are <Positives, Negatives, Sentiment> where
    - Positives: sum positive sentiment counts for each word in the tweet
    - Negatives sum positive sentiment counts for each word in the tweet
    - 1 for Positive and 0 for Negative

You don't have to build the NLP system from scratch. Modules, like the one found at https://github.com/necromuralist/Neurotic-Networking, can be utilized to help you in your project.

Data Files:

Test and Training data file format:

tab delimited

classification is 0/1 = negative / positive

Line schema: classification <tab> tweet

Trial data format:

tab delimited

Line schema: Username <tab> user screen name <tab> time <tab> is a retweet <tab> tweet text

What to turn in:

A zip file with:

(50pts) Documented .py file(s) to train and test your classifier, read in data files, and produce plots

(10pts) README file with instructions how to run your code

(40pts) A pdf file with two sections:

Section 1- description of your code/design for Bayesian classification; accuracy data demonstrating the performance on the testing/training data; analysis of what improved accuracy

Section 2- analysis of the additional tweets around the election demonstrating trends around the dates surrounding the election and/or geographic analysis of sentiment.