Sentiment Analysis by Voted Classifier on Live Twitter Data

Himanshu Gupta (hxg151930@utdallas.edu)
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas
Richardson, Texas-75080

Abstract:

In this paper, we examine the sentiment analysis on twitter data. The contributions of the paper are: (1) we introduce a Voted classifier which provides the polarity of a sentiment to be positive or negative along with Confidence factor on the decision of polarity of sentiment. (2) We have also build a module for with above functionality and used it on the twitter streaming data in order to produce a graph. This graph will help us to visualize the state or trend for product/company/new launch/hot topic. This will help us to know the exact market trend according to twitter comments.

1. Introduction:

Microblogging websites have evolved to become a source of varied kind of information. This is due to nature of microblogs on which people post real time messages about their opinions on a variety of topics, discuss current issues, complain, and express positive sentiment for products they use in daily life. In fact, companies manufacturing such products have started to poll these microblogs to get a sense of general sentiment for their product. Many times these companies study user reactions and reply to users on microblogs. One challenge is to build technology to detect an overall sentiment of the data based on a voted classifier from other classifiers. In this paper, we look at one such popular microblog called Twitter and build models for classifying "tweets" into positive and negative sentiment.

We build models for classifying sentiment into positive and negative classes. In order to classify the data we have used 6 different classifier, which are Naïve Bayes Classifier, Multinomial Naïve Bayes Classifier, Bernoulli naive Bayes classifier, Logistic Regression Classifier, Linear SVM classifier and Stochastic Gradient Descent Classifier. All the mentioned classifiers trained on well classified data. We then iterate through our list of classified objects, then, for each one, we ask it to classify based on the features. The classification is being treated as a vote. After we are done iterating, we then return the model (votes), which is just returning the most popular vote. We also added a parameter **confidence**. Since we are doing algorithms voting, we can also tally the votes for and against the winning vote, and call this "confidence." For example, 3/5 votes for positive is weaker than 5/5 votes.

We use manually annotated Twitter data for our experiments. One advantage of this data, over previously used data-sets, is that the tweets are collected in a streaming fashion and therefore represent a true sample of actual tweets in terms of language use and content. The work described above can be a module and is used on Twitter data for providing us a streaming graph to visualize the overall trend on the twitter for a particular topic or content. This type of analysis will help us to visualize an overall opinion on Twitter regarding the mentioned topic and content.

The rest of the paper is organized as follows. In section 2, we discuss the related work to this theory. In section 3, we give the literature survey for this paper. In section 4 we discuss experiment and results of theory. In section 5 will look forward to conclusion. We conclude and give future directions of research in section 6.

2. Related work:

Sentiment analysis is a growing area of Natural Language Processing with research ranging from document level classification (Pang and Lee 2008) to learning the polarity of words and phrases (e.g., (Hatzivassiloglou and McKeown 1997; Esuli and Sebastiani 2006)). Given the character limitations on tweets, classifying the sentiment of Twitter messages is most similar to sentence-level sentiment analysis (e.g., (Yu and Hatzivassiloglou 2003; Kim and Hovy 2004)); however, the informal and specialized language used in tweets, as well as the very nature of the microblogging domain make Twitter sentiment analysis is a very different task. It's an open question how well the features and techniques used on more well-formed data will transfer to the microblogging domain. In the past year there have been a number of papers looking at Twitter sentiment and buzz (Jansen et al. 2009; Pak and Paroubek 2010; O'Connor et al. 2010; Tumasjan et al. 2010; Bifet and Frank 2010; Barbosa and Feng 2010; Davidov, Tsur, and Rappoport2010).Other researchers have begun to explore the use of part-of-speech features but results remain mixed. Features common to microblogging (e.g., emoticons) are also common, but there has been littleinvestigationintotheusefulnessofexistingsentimentresources developed on non-microblogging data. Researchers have also begun to investigate various ways of automatically collecting training data. Several researchers rely on emoticons for defining their training data (Pak and Paroubek 2010; Bifet and Frank 2010). (Barbosa and Feng 2010) exploit existing Twitter sentiment sites for collecting training data. (Davidov, Tsur, and Rappoport 2010) also use hashtags for creating training data.

3. Literature Survey:

Dataset: For development and training purposes we have used dataset from twitter which has 5300+ positive and 5300+ negative movie reviews, which are much shorter. These should give us a bit more accuracy from the larger training set, as well as be more fitting for tweets from Twitter. The data is available from https://pythonprogramming.net/. For streaming data we have used tweepy (https://apps.twitter.com/)which is provided by the twitter for development and testing of the results of the experiments.

Construction of Voter Classifier: We build a module which provides the polarity of a given text as positive or negative comment. We have used 6 different classifier, which are Naïve Bayes Classifier, Multinomial Naïve Bayes Classifier, Bernoulli naive Bayes classifier, Logistic Regression Classifier, Linear SVM classifier and Stochastic Gradient Descent Classifier. All these classifiers provide the accuracy percentage and their vote as positive or negative for the given text.

Let us see the brief on each classifier:

- Naive Bayes Classifier: Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. The conditional distribution over the class variable C is:

  $$p(C_k|x_1,\dots,x_n) = \frac{1}{Z} p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

  where the evidence Z = p(\mathbf{x}) is a scaling factor dependent only on x_1, \dots, x_n, that is, a constant if the values of the feature variables are sknown.

- Multinomial Naïve Bayes Classifier : With a multinomial event model, samples (feature vectors) represent the frequencies with which certain events have been generated by a multinomial $(p_1, \dots, p_n)$ where p_i is the probability that event i occurs (or K such multinomial in the multiclass case). A feature vector $\mathbf{x} = (x_1, \ldots, x_n)$ is then a histogram, with $x_i$counting the number of times event i was observed in a particular instance. This is the event model typically used for document classification, with events representing the occurrence of a word in a single document. The likelihood of observing a histogram x is given by

$$p(\mathbf{x}|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_{ki}{}^{x_i}$$

- Bernoulli naive Bayes classifier: In the multivariate Bernoulli event model, features are independent Booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks,[9] where binary term occurrence features are used rather than term frequencies. If x_i is a Boolean expressing the occurrence or absence of the i'th term from the vocabulary, then the likelihood of a document given a class $C_k$is given by

$$p(\mathbf{x}|C_k) = \prod_{i=1}^{n} p_{ki}^{x_i}(1 - p_{ki})^{(1-x_i)}$$

  where p_{ki} is the probability of class $C_k$generating the term w_i. This event model is especially popular for classifying short texts. It has the benefit of explicitly modelling the absence of terms. Note that a naive Bayes classifier with a Bernoulli event model is not the same as a multinomial NB classifier with frequency counts truncated to one.

- Logistic Regression Classifier: logistic regression is a classification method that generalizes logistic regression to multiclass problems, i.e. with more than two possible discrete outcomes. That is, it is a model that is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables (which may be real-valued, binary-valued, categorical-valued, etc.).

- Linear SVM classifier: support vector machines are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier.

- Stochastic Gradient Descent Classifier: Stochastic gradient descent is a gradient descent optimization method for minimizing an objective function that is written as a sum of differentiable functions. Stochastic gradient descent is a popular algorithm for training a wide range of models in machine learning, including (linear) support vector machines, logistic regression (see, e.g., Vowpal Wabbit) and graphical models. When combined with the backpropagation algorithm, it is the de facto standard algorithm for training artificial neural networks.

All the mentioned classifiers trained on well classified data. We then iterate through our list of classified objects, then, for each one, we ask it to classify based on the features. The classification is being treated as a vote. After we are done iterating, we then return the model (votes), which is just returning the most popular vote.

Confidence: Since we are doing algorithms voting, we can also tally the votes for and against the winning vote, and call this "confidence." For example, 3/5 votes for positive is weaker than 5/5 votes. This will help us determining that for a particular sentiment how strong is the opinion about the polarity of text.

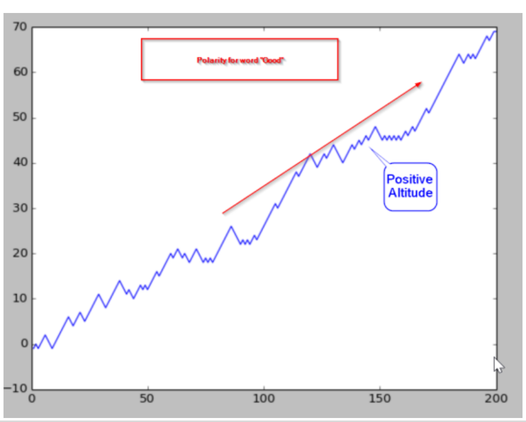
4. Experiment and Results:

We constructed a module which will take input text (or tweets). It will formulate the text with each classifier and based on the voting by each classifier it will return us the polarity of text and the confidence. This module is put to test with the twitter live streaming API (tweepy).

We have to input the topic or the content for which we want to survey and the tool will provide all the live tweets with their polarity and confidence. We will then draw a graph for particular content.

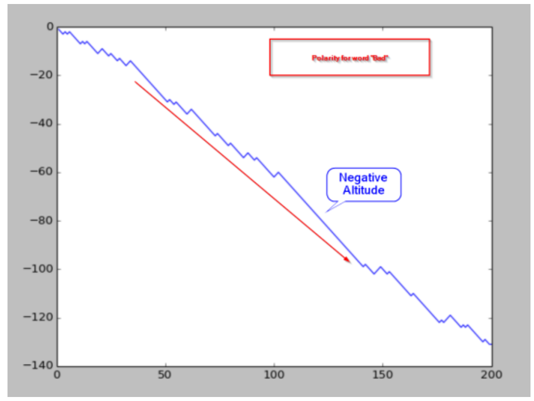
The graph is plotted on the basis of the polarity for each positive we are giving +1 in the graph and for each negative the graph will slope -1 in the graph. The curve obtained help us to determine whether the given content has positive impact on the twitter audience or it has the negative impact on the audience.

For testing the algorithm, let us take the first content to be a positive word say "good". We expect the resulting graph to show more positive altitude as the word is mostly used to annotate the positive comments.



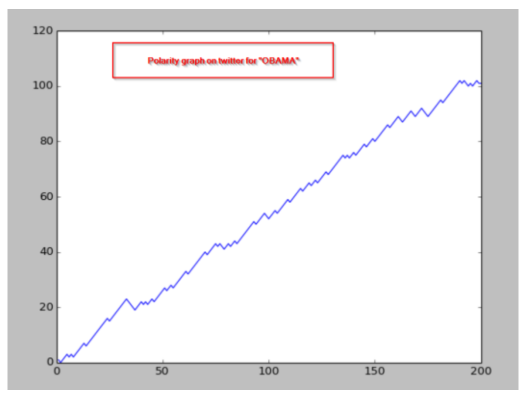
The result shows a positive altitude for the word "Good".

Simillarly we will test the result for a negative word say "Bad". We expect the resulting graph to show more negative altitude as the word is mostly used to annotate the negative comments.



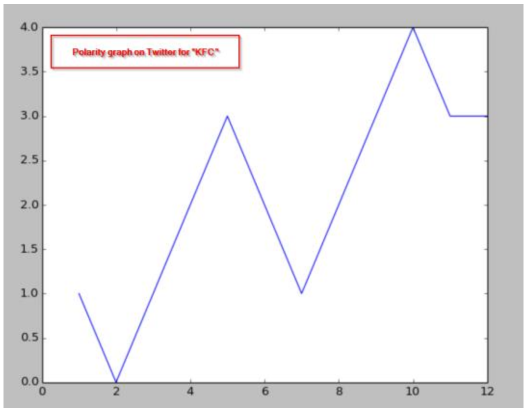
The result shows a negative altitude for the word "Bad".

Thus we can say that the trained algorithm is working well. We now put this to extract the polarity graph from twitter for a current hot topic. As the elections are around the corner, let us see what the twitter reactions about "Obama" is.



This shows that on twitter the president has mostly positive tweets about him and has a good reputation on twitter.

Similarly we can see the polarity graph for a particular brand say "KFC" and the reactions on the twitter can be plotted as



The graph shows mixed reviews of the people on twitter for brand KFC as we have positive and negative slopes on the graph.

5. Conclusions:

Our experiments on twitter sentiment analysis show that we are able to design a module which can be implemented on live twitter data or the static data to give us an overall polarity of the text. The confidence of the polarity from votes of various classifier help us to give a more accurate result for the input tweet or text.

The trained algorithm using the voter classifier provides the accurate result and thus can be used to survey the various hot topics on the twitter. This will help to obtain an overall polarity of a particular topic which can be useful for judging the overall reactions of public on twitter.

6. Future Work:

There are number of ways in which we can improve this work. We can look forward to measure the sentiment analysis of speech and provide a polarity for graph for a particular speech given by a person. Of course this will include lot of other factors and a proper training of algorithm need to be done as well. We can also look forward to comparison of the tweets or comments and can provide the confidence regarding the better polarity. This tool can be put to use with twitter website as well so that as soon as people comment they should know the polarity of the tweet they have done. We also need to remove the noisy data from the streaming tweets. The training of algorithm on larger and larger dataset will help this tool to improve more and provide the finer result as well. In the end, we would like to say that there can be a lot that can be done by using this type of module as it has lot of scope of improvisation and has lot of practical implementations that can be thought off for this.

7. References:

Barbosa, L., and Feng, J. 2010. Robust sentiment detection on twitter from biased and noisy data. In Proc. of Coling.

Bifet,A.,and Frank,E. 2010. Sentiment knowledge discovery in twitter streaming data. In Proc. of 13th International Conference on Discovery Science.

Kim, S.-M., and Hovy, E. 2004. Determining the sentiment of opinions. In Proceedings of Coling.

O'Connor, B.; Balasubramanyan, R.; Routledge, B.; and Smith,N.2010. From tweets to polls: Linking text sentiment to public opinion time series. In Proceedings of ICWSM.

Pak, A., and Paroubek, P. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In Proc. of LREC.

Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2):1–135.

Schapire, R. E., and Singer, Y. 2000. BoosTexter: A boosting-based system for text categorization. Machine Learning 39(2/3):135–168.

Tumasjan, A.; Sprenger, T. O.; Sandner, P.; and Welpe, I. 2010. Predicting elections with twitter: What 140 characters reveal about political