ISSN: 2321-7782 (Online)
Volume 3, Issue 3, March 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Machine Learning for Sentiment Analysis

Omkar Pradeep Acharya[1]
T.E. Computer Engineering
Pune Institute of Computer Technology
Pune, India

Parag Raj Ahivale[2]
T.E. Computer Engineering
Pune Institute of Computer Technology
Pune, India

Abstract: ***Through the proliferation of social media, nowadays, the technology encounters rapid development. The way in which people express their views has been changed. Micro-blogging has now become the most popular way of doing this and Twitter is widely used for this. As everyone uses blogging for expressing his/her views, it has got a vast amount of sentiment rich data in form of tweets, status updates, blog posts, etc. The sentiment analysis of Twitter is little a bit difficult due to its maximum characters limit which is 140 per tweet. Analysing user sentiments towards electronic products through their review comments and ratings can be economically profitable to product developers. There are two approaches for having a sentiment analysis on tweets: Knowledge based approach and Machine Learning Techniques. In this study, we are mainly focusing on ML approaches by using classifiers like Naïve Bayes, Maximum Entropy. A feature vector is proposed for classifying the tweets as positive, negative and extract peoples' opinion about electronic products like mobiles and tablets.***

Keywords: Twitter, Sentiment Analysis, Machine Learning Techniques, Naïve Bayes, Maximum Entropy

I. INTRODUCTION

Today, people have changed their way of expressing views as most of them are using micro-blogging. Micro-blogging sites have got a huge amount of sentiment rich data in form of tweets, status updates, blog posts, etc. Analyzing user sentiments towards electronic products through their review comments and ratings can be economically profitable to product developers. There are two basic approaches by which sentiment analysis can be performed; *Knowledge Based Techniques* and *Machine Learning Techniques*. Knowledge Based Techniques, also known as Symbolic Techniques, require a large amount of data of predefined emoticons to identify the sentiments whereas Machine Learning Techniques simplify this overhead by making use of training set to develop an efficient sentiment classifier which classifies the sentiments. As some predefined dataset is not required, Machine Learning Techniques make the life easy for sentiment analysis.

Sentiment Analysis can be classified according to different levels viz. coarse level sentiment analysis, fine level sentiment analysis. Coarse level sentiment analysis deals with determining the analysis of the entire document, on the other hand, Fine level sentiment analysis takes into consideration the sentiment of attributes. We, specially, have chosen Twitter as the source of sentiment, because of two main reasons. First, the maximum character limit of Twitter per tweet is very less i.e. 140 characters per tweet. Second, due to this limit, sentiment analysis of Twitter makes it different as well as difficult as compared to other sources of sentiments. Machine Learning techniques that we have used in this paper include Naïve Bayes and Maximum Entropy. In the end, we have compared all these approaches on the basis of their accuracy. Accordingly, we are classifying all the available sentiments into two categories: positive and negative.

II. DATA CLASSIFICATION PROCESS

» ***Learning:***

Training data are analysed by a classification algorithm. Here, the class label attribute is sentiment, and the learned model or classifier is represented in the form of classification rules.

» ***Classification:***

Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples. If we were to use the training set to measure the accuracy of the classifier, this estimate would likely be optimistic, because the classifier tends to over-fit the data.

The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier, because the class label of each training tuple is provided, this step is also known as 'supervised learning'.

III. Related Work

» Knowledge Based Approaches:

Turney (2002) used bag of words approach. Sentiment of each of the word from document is determined and then by using aggregation function, overall sentiment of the document is determined.

SemEval (2007) used coarse grained and fine grained approach. Binary classification of news was done in coarse grained approach and Different level classification was done in fine grained approach.

» Machine Learning Approaches:

Domingos (2006) used Naïve Bayes classifier. The result was found as Naïve Bayes Classifier works well for certain problems with highly dependent features. This was surprising as Naïve Bayes uses Conditional Independence in which features are considered as independent of each other.

Xia (2011) used Ensemble Classifier. Naive Bayes, Maximum Entropy and Support Vector Machines are selected as base classifiers. They applied different ensemble methods like fixed combination, weighted combination and Meta-classifier combination for sentiment classification and obtained better accuracy.

IV. Proposed Solution

Tweets are nothing but short texts may or may not be containing slang words, misspellings and twitter specific attributes like emoticons, hash-tags, etc. Sentiment analysis of these tweets can be done in four major steps:

1. Dataset Creation
2. Pre-processing of Tweets
3. Creation of Feature Vector
4. Sentiment Classification

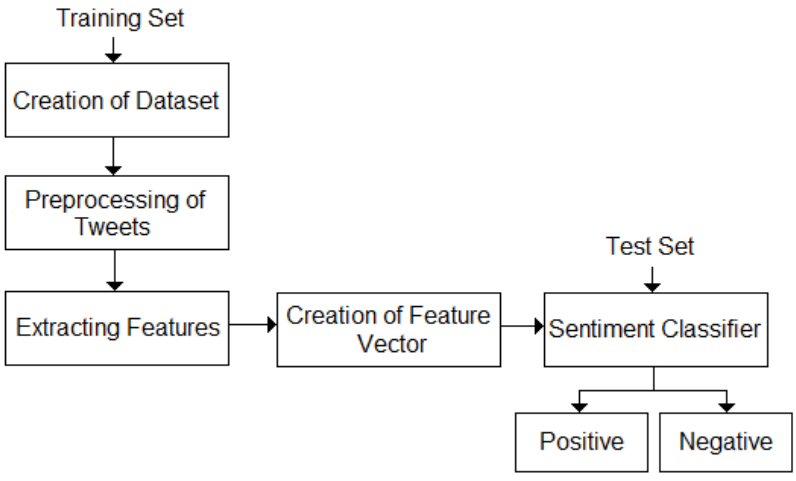


Fig1. Steps for Sentiment Analysis by Machine Learning Approach

a) Dataset Creation:

TABLE I

STATISTICS OF THE DATASET

Dataset	Positive	Negative	Total
Training Set	2000	2000	4000
Test Set	25	15	40

Dataset of tweets can be created by collecting tweets over a period of a month or two from Twitter. We have created a dataset by collecting tweets from Twitter API which contains some positive and some negative tweets.

b) *Pre-processing of Tweets:*

1. Lower Case: Convert a tweet to lower case
2. URLs: Replace all URLs with a generic URL
3. Username: Convert all usernames like @username into the same word AT_USERNAME
4. Hash-tag: Remove # from the start to make it a plain text. e.g. #twitter to twitter
5. Punctuation and White Spaces: Remove all the additional white spaces along with punctuations to make a plain text from tweet e.g. “That is too expensive!” into “that is too expensive”

c) Feature Vector Creation:

Part of speech (POS) tagging is done by classifying all the word from respective tweet into word classes as nouns, verbs, adjectives, adverbs, etc. It is obvious that adjectives, adverbs and verbs have more relevance in determining sentiment. POS Tagger can be used for this tagging purpose which gives the analysis result. Python uses NLTK package which provides a function pos_tag (*tweet_goes_here*) which performs tagging.

Special keyword is given a weight of ‘1’ if it is positive and ‘-1’ if it is negative. Negative emoticons have a great impact on the sentiment of overall sentence, so emoticon is included as a separate feature. Total count of positive keywords, negative keywords, positive hast-tags and negative hash-tags are included as features in feature vector.

Feature vector that we have considered in this analysis contains 8 relevant features:

» Part of Speech (POS) tag
» Special keyword
» Presence of negation
» Emoticon
» Number of positive keywords
» Number of negative keywords
» Number of positive hash-tags
» Number of negative hash-tags

d) Sentiment Classification:

After creating feature vectors, classification techniques like Naïve Bayes, Maximum Entropy can be used for sentiment classification.

V. CLASSIFICATION TECHNIQUES

a) Naïve Bayes Classifier:

This classifier assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered "naïve." Let H be the hypothesis that the data tuple **X** belongs to a specific class C. Now, we want to determine P (H | **X**), the probability that the hypothesis H holds given the data tuple **X**. In other words, we are looking for the probability that tuple **X** belongs to class C given that we know the attribute description of **X**.

P (H | **X**) is posterior probability of H conditioned on **X**. P (H) is the prior probability of H. P (**X**) is the prior probability of **X**. Bayes' theorem P (H | **X**) = P (**X** | H) * P (H) / P (**X**)

This classifier predicts that the tuple **X** belongs to a class C*i* if and only if P (C*i* | **X**) > P (C*j* | **X**).

Thus we have to maximize P (C*i* | **X**). The class *Ci* for which *P* (*Ci* | **X**) is maximized is called the *maximum posteriori hypothesis.* Thus Bayes' theorem changes as follows: P (C*i* | **X**) = P (**X** | C*i*) * P (C*i*) / P (**X**)

As this classifier assumes class conditional independence as discussed earlier, we can state that:

$$P(\mathbf{X} \mid Ci) = \prod_{k=1}^{n} P(xk \mid Ci)$$

$$P(\mathbf{X} \mid Ci) = P(x1 \mid Ci) * P(x2 \mid Ci) * P(x3 \mid Ci) * \ldots * P(xn \mid Ci)$$

Where **X** = {*x1, x2… xn*} Here, *x1, x2… xn* are independent features like number of positive and negative keywords, count of positive and negative hash-tags, etc. from feature vector .

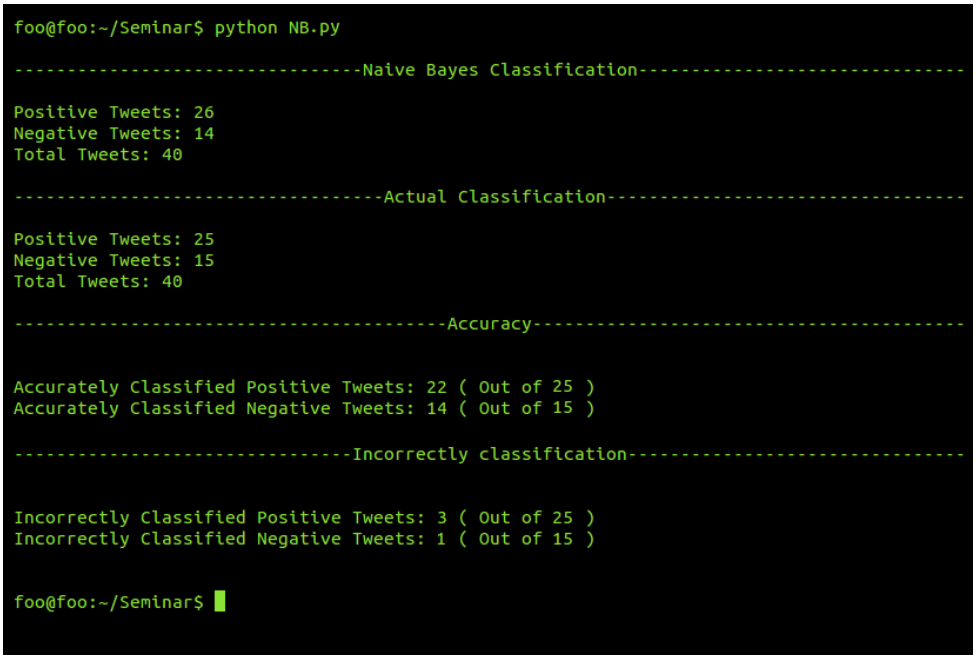


Fig.2 Accuracy Measure of Naïve Bayes Classifier

b) Maximum Entropy:

The Maximum Entropy classifier is a probabilistic classifier which belongs to the class of exponential models. Unlike the Naïve Bayes classifier, the Max Entropy does not assume that the features are conditionally independent of each other. The MaxEnt is based on the Principle of Maximum Entropy and from all the models that fit our training data, selects the one which has the largest entropy. The Max Entropy classifier can be used to solve a large variety of text classification problems such as language detection, topic classification, sentiment analysis and more.

Due to the minimum assumptions that the Maximum Entropy classifier makes, we regularly use it when we don't know anything about the prior distributions and when it is unsafe to make any such assumptions. Moreover Maximum Entropy classifier is used when we can't assume the conditional independence of the features. This is particularly true in Text Classification problems where our features are usually words which obviously are not independent. The Max Entropy requires more time to train comparing to Naive Bayes, primarily due to the optimization problem that needs to be solved in order to estimate the parameters of the model. Nevertheless, after computing these parameters, the method provides robust results and it is competitive in terms of CPU and memory consumption.

In Maximum Entropy Classifier, no assumptions are taken regarding the relationship between features. This classifier always tries to maximize the entropy of the system by estimating the conditional distribution of the class label. The conditional distribution is defined as:

$$P\lambda\,(y|X) = 1/Z(X)\ \exp\{\sum \lambda i * fi(X, y)\}$$

'X' is the feature vector and 'y' is the class label. Z(X) is the normalization factor and λ i is the weight coefficient. fi(X, y) is the feature function which is defined as

$$fi(X, y) = \{ \quad 1, \quad X=xi \text{ and } y = yi$$

$$0, \quad \text{otherwise} \quad \}$$

VI. Performance Analysis

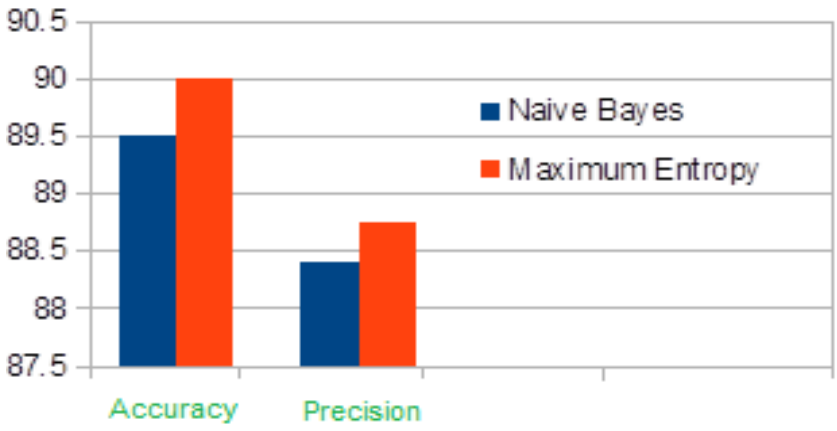


Fig.3 Performance of Different classifiers in Twitter Sentiment Analysis

Classification algorithms like Naïve Bayes, Maximum Entropy can be compared on the basis of following factors:

» *Accuracy:* This refers to the ability of a classifier to correctly predict the class label of new or previously unseen data.

» *Speed:* This refers to the computational costs involved in generating and using the given classifier.

» *Robustness*: This is the ability of the classifier to make correct predictions given noisy data or data with missing values.

» *Scalability:* This refers to the ability to construct the classifier efficiently given large amounts of data.

VII. Conclusion

Sentiment analysis of tweets can be done in two ways: Knowledge based techniques and Machine Learning techniques. Machine Learning techniques find themselves very easy and efficient as compared to Knowledge based techniques. To perform sentiment analysis, first the dataset containing both positive and negative tweets is created. Then slang words and misspellings are handled which makes sentiment analysis of twitter very difficult. The feature vector is created by using two steps extraction after preprocessing to classify the sentiments as positive and negative. Classification techniques involved are Naïve Bayes and

Maximum Entropy. If they are compared according to their accuracy and precision, then we found out that almost all of them give the same result approximately.

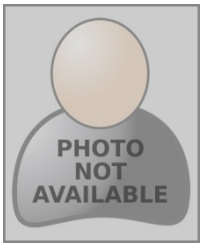
ACKNOWLEDGMENT

As students of Stanford's CS-229 MOOC, we would like to thank Coursera for providing such quality material online and our project guide Prof. S. S. Sonawane for her valuable inputs.

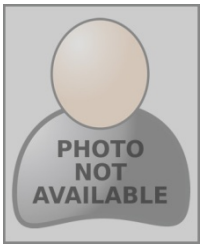
References

1. Sentiment Analysis in Twitter using Machine Learning Techniques (IEEE – 31661, 4th ICCCNT 2013, July 4 - 6, 2013, Tiruchengode, India)
2. Micro blogging Sentiment Analysis with Lexical Based and Machine Learning (International Conference of Information Technology, 2013)
3. Thumbs up? Sentiment Classification using Machine Learning Techniques
4. Social Media Analysis for Product Safety using Text Mining and Sentiment Analysis (978-1-4799-5538-1/14/$31.00 ©2014 IEEE)
5. Stanford's CS-229 Machine Learning by Andrew Ng

AUTHOR(S) PROFILE



Omkar Pradeep Acharya, is 3rd year Computer Engineering student from Pune Institute of Computer Technology. His area of interests includes Machine Learning, Sentiment Analysis, Data Mining, Web Design and Object Oriented Programming.



Parag Raj Ahivale, is 3rd year Computer Engineering student from Pune Institute of Computer Technology. His area of interests includes Machine Learning, Business Intelligence, Data Mining and Object Oriented Programming.

