Twitter Sentiment Analysis Using Machine Learning Algorithm with Python

L.Prasanna Kumar[1], Sampathirao Suneetha[2]
[1]Assoc.Prof, Department of CSE, Dadi Institute of Engineering & Technology, Visakhapatnam.
[2]Research scholar, AUCE(A), Andhra University, Visakhapatnam.

ABSTRACT: *From the last few years, use of social networking sites has been increased rapidly. Nowadays, large amount of data is generated by social networking sites. Billions of people can express their feelings and thoughts on verity of topics via micro blogging websites such as Twitter, Scoop.it, Pinterest, etc. In this paper, we will discuss the extraction of sentiment from a famous micro blogging website, Twitter where the user posts their views and opinion.Sentiment analysis on twitter data helps in providing some prediction on business intelligence. Here We use naive Bayesian algorithm for processing Twitter data set that is available on the twitter website in the form of reviews, feedback, and comments. Results of sentiment analysis on twitter data will be displayed as different sections presenting positive and negative sentiments.*
Key Words*: Twitter, Sentiment Analysis, Feature extraction, Natural Language Toolkit, Naïve Bayes classifier, Sentiment polarity.*

I. INTRODUCTION:

Comments, reviews and ratings of the people play an important role in social media to determine whether a population is satisfied with the particular services, products. It helps in predicting the sentiment of a wide variety of people on a particular event like the review of a movie, their opinion on various topics spread around the world. These data are essential for sentiment analysis [2]. In the process of discovering the overall sentiment of population and retrieval of data from sources like Twitter, Facebook, Blogs are very important and essential. Twitter generates huge data that cannot be handled manually to extract some useful information and therefore, the ingredients of automatic classification are required to handle those data. The Twitter interface allows the user to post short messages and that can be read by any other Twitter user. Twitter contains a variety of text posts and grows every day (21 million tweets per hour, as measured in 2015, hence there is a need to automate the process of sentiment analysis so as to ease the tasks of determining public's opinions without reading millions of tweets manually. This process of analyzing the opinions expressed in these huge opinionated user generated data is usually called Sentiment Analysis or Opinion Mining which is a very interesting research in present days.

Sentiment analysis is a process of automatically identifying opinions expressed in text format which are either positive, negative or neutral opinion about an entity (i.e. product, people, topic, event etc).Sentiment classification can be done at Document level, Sentence level and Aspect or Feature level [1]. In Document level the whole document is used as a basic information unit to classify it either into positive or negative class. Sentence level sentiment classification classifies each sentence first as subjective or objective and then classifies into positive, negative or neutral class. There is no much difference between the above two methods as sentence is just a short document. Aspect level sentiment classification deals with identifying and extracting product features from the source data [1].

The organization of paper is as follows: Section 1 gives an overview about various sentiment classification techniques. Section 2 explains the objective of project. Section 3 gives literature survey of sentiment analysis. Section 4 explains various steps needed for sentiment analysis using machine learning. Section 5 provides an outline of Naïve Bayes classifier for sentiment classification.

II. OBJECTIVE:

- The main objective of this project is to show how sentimental analysis can help improve the user experience over a social network.
- The learning algorithm will learn what our emotions are from statistical data then perform sentiment analysis. Our main objective is also maintaining accuracy in the final result.
- The main goal of such a sentiment analysis is to discover how the audience perceives the television show.
- The Twitter data that is collected will be classified into two categories; positive or negative. An analysis will then be performed on the classified data to investigate what percentage of the audience sample falls into each category.
- Particular emphasis is placed on evaluating different machine learning algorithms for the task of twitter sentiment analysis.

III. LITERATURE SURVEY: Sentiment classification aims to automatically predict sentiment polarity (e.g., positive or negative) of users publishing sentiment data (e.g., reviews, blogs). Although traditional classification algorithms can be used to train sentiment classifiers from manually labeled text data, the labeling work can be time-consuming and expensive. Meanwhile, users often use some different words when they express sentiment in different domains. If we directly apply a classifier trained in one domain to other domains, the performance will be very low due to the differences between these domains. In this work, we develop a general solution to sentiment classification when we do not have any labels in a target domain but have some labeled data in a different domain, regarded as source domain II] A sentiment classification method that is applicable when we do not have any labeled data for a target domain but have some labeled data for multiple other domains, designated as the source domains. We automatically create a sentiment sensitive thesaurus using both labeled and unlabeled data from multiple source domains to find the association between words that express similar sentiments in different domains. The created thesaurus is then used to expand feature vectors to train a binary classifier. Unlike previous cross domain sentiment classification methods, our method can efficiently learn from multiple source domains. Our method significantly outperforms numerous baselines and returns results that are better than or com-parable to previous cross-domain sentiment classification methods on a benchmark dataset containing Amazon user reviews for different types of products.] Merchants selling products on the Web often ask their customers to review the products that they have purchased and the associated services. As e-commerce is becoming more and more popular, the number of customer reviews that a product receives grows rapidly. For a popular product, the number of reviews can be in hundreds or even thousands. This makes it difficult for a potential customer to read them to make an informed decision on whether to purchase the product. It also makes it difficult for the manufacturer of the product to keep track and to manage customer opinions. For the manufacturer, there are additional difficulties because many merchant sites may sell the same product and the manufacturer normally produces many kinds of products. In this research, we aim to mine and to summarize all the customer reviews of a product. This summarization task is different from traditional text summarization because we only mine the features of the product on which the customers have expressed their opinions and whether the opinions are positive or negative. We do not summarize the reviews by selecting a subset or rewrite some of the original sentences from the reviews to capture the main points as in the classic text summarization.

IV. PROPOSED SYSTEM:

The general steps take to complete this project are:

1. Get twitter data.
2. Preprocess my tweets so that no capitalization, punctuation, or non ASCII characters are present, as well as splitting the tweet into an array holding each word in a separate holder
3. Create a bag of common words that appear in my tweets
4. Create a frequency table of words that have positive and negative hits
5. Test my frequency table by using test sentences using Sentiment analysis algorithm.

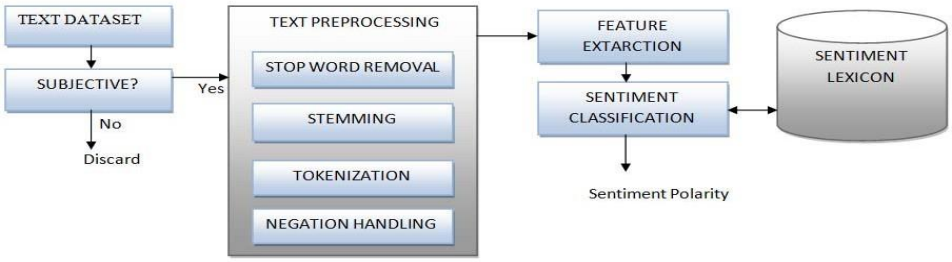

Fig1: Sentiment Analysis Process

Text Preprocessing:

1) Stop word removal: Pronouns (he/she, it), articles (a, the), prepositions (in, near, beside) are stop words. They provide no or little information about sentiments. There is a list of stop words available on the internet. It can be used to remove them in the pre-processing step.
2) Stemming: It is the process of removing prefixes and suffixes. For example 'playing', 'played' can be stemmed to 'play'. It helps in classification but sometimes leads to decrease in classification accuracy.
3) Tokenization: Here the sentences are divided into words or tokens by removing white spaces and other symbols or special characters.

4) Negation handling: Negation words like 'not' inverts the meaning of whole sentence. For example: The movie was not good has 'good' in it which is positive but 'not' inverts the polarity to negative.
5) Punctuation Removal: Punctuation marks such as comma or colon often carry no meaning for the textual analysis hence they can be removed from input text.

Feature Extraction:

1) N grams- n grams refers to consecutive n terms in text. One can take only one word at a time (unigram) or two words (bigram) up to n accordingly. Some sentiments can't be captured with unigram feature. For example this drink will knock your socks off. It is a positive comment if socks off is taken together and negative in case of only unigram model (off).
2) Parts of speech tagging- It is a way toward denoting a word in a content as comparing to parts of speech in light of both its definition and its association with contiguous words. Nouns, pronouns, adjectives, adverbs etc are examples parts of speech. Adjectives and adverbs hold most of the sentiments in text.

Sentiment classification:

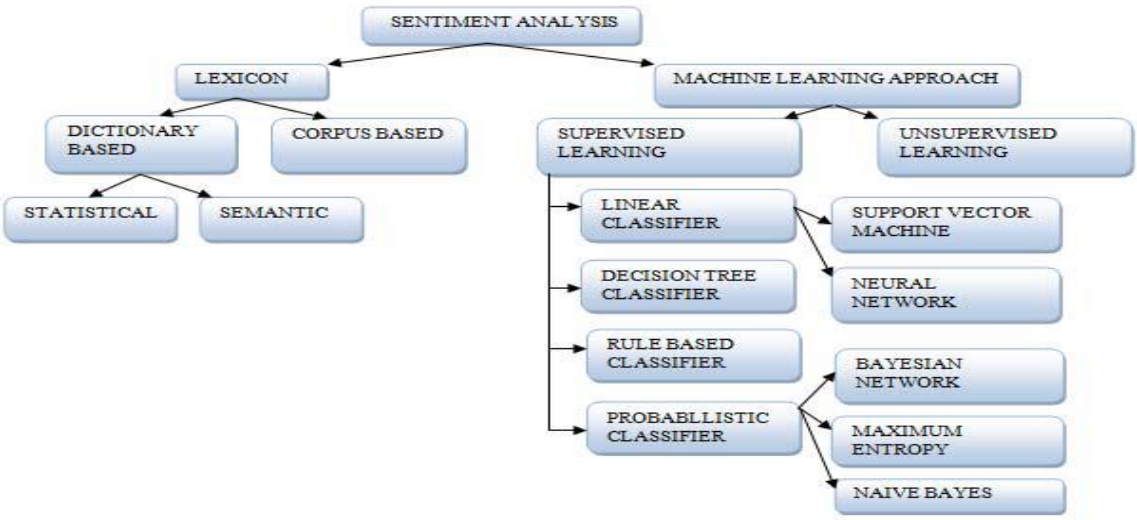

Fig2: Classification of Sentiment Analysis

Two approaches are mainly used

1) Subjective lexicon: Subjective lexicons are collection of words where each word has a score indicating the positive, negative, neutral and objective nature of text. In this approach, for a given piece of text, aggregation of scores of subjective words is performed i.e. positive, negative, neutral and objective word scores are summed up separately. In the end there are four scores. Highest score gives the overall polarity of the text.[3]

a) Dictionary based approach- In this approach a set of opinion words are manually collected and a seed list is prepared. Then we search for dictionaries and thesaurus to find synonyms and antonyms of text. The newly found synonyms are added to the seed list. This process continues until no new words are found.

Disadvantage: difficulty in finding context or domain oriented opinion words

b) Corpus based approach- Corpus is collection of writings, often on a specific topic. In this approach, seed list is prepared and is expanded with the help of corpus text.[4] Thus is solves the problem of limited domain oriented text. It can be done in two ways

2) Machine learning: This is an automatic classification technique. Classification is performed using text features. Features are extracted from text. It is of two types- supervised and unsupervised.

a) Supervised Learning

The majority of real time machine learning uses supervised learning approaches.

Supervised learning a pre defined model where we have one input variables (x) and an output variable (Y) and we use an algorithm to learn the mapping function from the input to the output.

$$Y = f(X)$$

The main air is to estimate the mapping function so that when we have new input data (x) that we can predict the output variables (Y) for that data.

The most widely used Supervised learning algorithms are: Support Vector Machines, linear regression, logistic regression, naive Bayes, ect..

b) Unsupervised Machine Learning

Unsupervised learning is process where we only have input data (X) and no respective output variables.
The main aim of unsupervised learning approach is to model the underlying structure or distribution in the data in order to learn more about the data.
Some of the most common algorithms used in unsupervised learning include: Clustering, Anomaly detection, Neural Networks etc.

V. MATHEMATICAL MODEL

Let S be the system that describes the tweet extraction, Preprocessing, Sentiment labeling, Sentiment AnalysisS= {Tw, Pt, Sl}
Tw =Tweets extracted from Twitter.
Sl={Pv, Nv}
Pv= {P1, P2,…,Pn}= Positive Class
Nv={ N1,N2,…,Nn }= Negative Class
Where, S= Sentimental analysis system. Pt =Pre-processing of Tweets (Slang word translation, Non-English word removal, PoS tagging, URL and Stop word removal). Sl=Sentiment Labeling using Sent Strength and Twitter Sentiment sentiment analysis tools (SVM to give more efficient and accurate results). P1, p2..Pn positive tweets collection class N1, N2...Nn Negative tweets collection class.

VI. Naïve Bayesian working methodology:

The Naïve Bayes classifier algorithm is the simplest and most frequently used classifier. Naïve Bayes classification algorithm computes the posterior probability of a label(class), based on the distribution of the words in the document. The model works with the Bag of Words feature extraction which ignores the position of the word in the document. It uses Bayes classification algorithm to predict the probability that a given feature set belongs to a particular label.

$$P(label/features) = \frac{P(label)*P(features/label)}{P(features)}$$

Given assumption states that all features are independent, the equation could be rewritten as follows:

P(label/features)=(P(label)*P(f1/label)*……*P(fn/label)) / P(features)

- P(label/features) is the posterior probability of class (c, target) given predictor (x, attributes).
- P(label) is the prior probability of class.
- P(features/label) is the likelihood which is the probability of predictor given class.
- P(features) is the prior probability of predictor.

Formula used for Algorithm:

$$\phi_{k|label=y} = P(x_j = k \mid label = y)$$

$$\phi_{k|label=y} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \text{ and } label^{(i)} = y\} + 1}{(\sum_{i=1}^{m} 1\{label^{(i)} = y\} n_i) + |V|}$$

$\phi_{k|label=y}$ = probability that a particular word in document of label(neg/pos) = y will be the k^{th} word in the dictionary.

m = Number of words in i^{th} document.

n_i = Total Number of documents.

Training:

Here, We have to generate training data and find the probability of occurrence in positive/negative train data files.
Calculate
ϕk/label = y for each label.
Calculate ϕk/label = y for each dictionary words and store the result (Here: label will be negative and positive).
The following tables and calculations shows detailed explanation of tweet data processing, feature extraction(BOW), analysis and tweet polarity classification based on Naïve Bayes Classifier.

DOC	TEXT	CLASS
1	I loved the movies	+
2	I hated the movies	-
3	A great movie. good movies	+
4	Poor acting	-
5	Great acting. a good movies	+

Ten Unique words: <I, love, the, movie, hate, a, great, poor, act, good>
Convert the document into Bag of words, where the attributes are possible words, and the values are the number of times a word occurs in the given document

DOC	I	loved	The	Movies	hated	a	great	poor	acti ng	good	CLASS
1	1	1	1	1							+
2	1		1	1	1						-
3				2		1	1			1	+
4								1	1		-
5				1		1	1		1	1	+

Documents with positive outcomes.

DOC	I	loved	the	movies	hated	a	great	poor	acting	good	CL ASS
1	1	1	1	1							+
3				2		1	1			1	+
5				1		1	1		1	1	+

P (+) =3/5=0.6
Compute:
p (i|+); p (love|+); p (the|+); p (movie|+); P (a|+); p (great|+); p (act|+); p (good|+)
Le n be the number of words in the (+) case and n_k the number of times word k occurs in these cases(+)

$$\text{Let } P(w_k/+) = \frac{(n_k+1)}{2n+|Vocabulary|}$$

Now, let's look at the negative examples

DOC	I	loved	the	Movies	hated	a	great	poor	acting	good	CLASS
2	1		1	1	1						-
4								1	1		-

$P(+)=3/5=0.6;\ p(w_k/+)=\frac{nk+1}{2n+|Vocabulary|}$

$P(-)=2/5=0.4$

$P(i|-)=\frac{1+1}{6+10}=0.125;$ $P(loved|-)=\frac{0+1}{6+10}=0.0625;$

$P(the|-)=\frac{1+1}{6+10}=0.125;$ $P(movies|-)=\frac{1+1}{6+10}=0.125;$

$P(a|-)=\frac{0+1}{6+10}=0.0625;$ $P(great|-)=\frac{0+1}{6+10}=0.0625;$

$P(acting|-)=\frac{1+1}{6+10}=0.125;$ $P(good|-)=\frac{0+1}{6+10}=0.0625;$

$P(hated|-)=\frac{1+1}{6+10}=0.125;$ $P(poor|-)=\frac{1+1}{6+10}=0.125;$

$P(i|+)=\frac{1+1}{14+10}=0.0833;$ $P(loved|+)=\frac{1+1}{14+10}=0.0833;$

$P(the|+)=\frac{1+1}{14+10}=0.0833;$ $P(movies|+)=\frac{5+1}{14+10}=0.2083;$

$P(a|+)=\frac{2+1}{14+10}=0.125;$ $P(great|+)=\frac{2+1}{14+10}=0.125;$

$P(acting|+)=\frac{1+1}{14+10}=0.0833;$ $P(good|+)=\frac{2+1}{14+10}=0.125;$

$P(hated|+)=\frac{0+1}{14+10}=0.0417;$ $P(poor|+)=\frac{0+1}{14+10}=0.0417;$

Now that we've trained our classifier,

Testing:

Let's classify a new sentence according to:

T1=log P(x| label= pos)+log P(label=pos)

Similarly calculate

T2=log P(x| label= neg)+log P(label=neg)

Compare T1&T2 to compute whether it has Negative or Positive sentiment

"I hated the poor acting"

If V_j= +; p(+)p(i|+)p(hate|+)p(the|+)p(poor|+)p(act|+)=6.03*10-7

If V_j= --; p(-)p(i|-)p(hate|-)p(the|-)p(poor|-)p(act|-)=1.22*10-5

Conclusion:

In this paper we have used Naïve Bayesian algorithm to calculate opinions of text for the review of the product. Here we have used positive words and negative words for sentiment analysis. We have twitter data for sentiment analysis. Further we extent this model using SVM for classification of opinion mining.

References:

[1]. Mr. S. M. Vohra, 2 Prof. J. B. Teraiya," A Comparative Study Of Sentiment Analysis Techniques", Journal Of Information, Knowledge And Research In Computer Engineering Issn: 0975 – 6760| Nov 12 To Oct 13 | Volume – 02, Issue – 02 Pg 313-317

[2]. F. Neri, C. Aliprandi, F. Capeci, M. Cuadros, T.By, "Sentiment Analysis on Social Media", IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2012, pp. 919 - 926

[3]. Kang Hanhoon, Yoo Seong Joon, Han Dongil., "Senti-lexicon and improved Naı¨ve Bayes algorithms for sentiment analysis of restaurant reviews", Expert Syst Appl ,39:6000–10, 2012

[4]. Keshtkar Fazel, Inkpen Diana., "A bootstraping method for extracting paraphrases of emotion expressions from texts" Comput Intell;vol. 0, 2012.

[5]. Nehal Mamgain, Ekta Mehta, Ankush Mittal and Gaurav Bhatt, "Sentiment Analysis of Top Colleges in India Using Twitter Data", (IEEE) ISBN -978-1-5090-0082-1, 2016.

[6] Halima Banu S and S Chitrakala, "Trending Topic Analysis Using Novel Sub Topic Detection Model", (IEEE) ISBN-978-1-4673-9745-2, 2016.

[7] Shi Yuan, Junjie Wu, Lihong Wang and Qing Wang, "A Hybrid Method for Multi-class Sentiment Analysis of Micro-blogs", ISBN- 978-1-5090-2842-9, 2016.