ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

Sentiment Analysis of Twitter Data from Political Domain Using Machine Learning Techniques

AmrutaTarlekar, Prof. Kodmelwar M.K

PG Student, Dept. of Computer Engineering, TSSM's BSCOER, Narhe, Pune, India

Assistant Professor, Dept. of Computer Engineering, TSSM's BSCOER, Narhe, Pune, India

ABSTRACT: In this paper, we design & develop a system called automatic sentiment analysis based on machine learning technique. Nowadays, people are enjoying social media online web services like twitter for expressing their opinions toward product bought and sentiment toward any person/political parties. Such people generated large volume of data or online reviews can be very useful for assessing the knowledge to political parties and even to public to know about the current scenario of politics within that region. However, it can be very challenging task for analyzing such huge amount of data and involves human interpretation. In this project, we developed a workflow to integrate both large-scale data mining techniques as well as qualitative analysis. We focused on public's Twitter posts, different micro blogging websites where public post their opinions/reviews about political leaders/parties to understand issues and problems that they have with them (political parties/leaders). We have conducted a qualitative analysis on samples taken from micro blogs/tweets related to political leaders/parties to identify different sentiments that is positive as well as negative aspects of public. Based on these results, we have implemented a multi-label classification algorithm to classify tweets/micro blogs. Reflecting public's reviews about particular political leader/party and we used this algorithm to train detector which will automatically detect sentiments (happy, sad, disgusting, and angry) from tweets and micro blogs. Hence, sentiment analysis or opinion mining aims to use automated tools to detect positive/negative aspects of opinion.

KEYWORDS: Sentiment analysis; data mining; social media; machine learning; Positive/Negative aspects of data.

I. INTRODUCTION

As of late, there has been a rapid of change in web services, internet technology, various types of social media sites such as discussion forums, micro blogs, and peer-to peer networks provides a affluent of information as well as posting online opinions/reviews about particular person or product has tremendously became a popular way for sharing their opinions or thoughts about particular political party/politician or services.

The social networking site Twitter will be the targeted social media site for this paper. Nowadays, billions of users are using the twitter can be used as rich source of data for mining information. There is an interface for retrieving the twitter data known as twitter corpus which provides free information in the form of a stream. Analysis of this information has led to a variety of research. Examples include prediction of elections and the stock market, notification of events such as earthquakes, analysis of natural disasters and public health information, estimation ofpublic sentiment during elections and recession.Such a user generated data can be very useful in assessing the general public's opinions, sentiment, and repercussion towards politician, political parties, products and services.

Recently conducted surveys have been revealed that such online reviews from public has played very important role in strategies of political parties & e-commerce companies. The information posted on such platforms is a huge amount of resource for obtaining the sentiment of the general public. The retrieval and analysis of such information is often referred to as sentiment analysis or opinion mining. Lots of new technologies are coming out in market for automatic sentiment analysis and to dig out hidden knowledge from such a huge amount of user-generated data on

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

social media.Sentiment analysis is becoming a popular way to classify the semantic orientation from the text; mainly because of social media sites which include online users who are free to express their feelings, thoughts and impressions concerning a specific topic. Recent search technologies can effectively help users to obtain a result data, which is related to their searched keywords. But, the semantic orientation of the content, which is much more important context in the reviews or opinions, is not populated by current search engines. For example, search engine like Google will give you around 78, 60,000 results for the query "reviews on Bhartiya Janta Party". If the current search engines can provide the semantic orientations in opinions or sentiments then it will be more useful for the public to decide right candidate for serving country. For the above mentioned political domain query may yield such report as "There are 10 000 results, of which 80% are thumbs up and 20% are thumbs down". To implement this type of technology requires the capability of discovering the negative as well as positive aspects of review. To find out whether a given review on particular politician or political party is in favour of or not supporting is similar to the traditional binary-classification problem. For given review, the classifier tries to classify the review into positive category or negative category. However, reviews in natural language are usually expressed in subtle or complex ways as to difficult to analyze. So, the challenge of classification may not be overcome by simple text-categorization approaches such as *n*-gram or keyword identification methods.

Hence, a better tool for automated classification of opinions is sentiment analysis using machine learning techniques into positive / negative aspects or polarity of public opinions.communication.

A. *Overview of Sentiment:*

Sentiment analysis involved in the study of opinion mining. Sentiments are defined as "an acquired and relatively permanent major neuropsychic disposition to react emotionally, cognitively, and co natively toward a certain object (or situation) in a certain stable fashion, with awareness of the object and the manner of reacting." Gordon [3] has a similar definition; sentiments are "socially constructed patterns of sensations, expressive gestures, and cultural meanings organized around a relationship to a social object, usually another person or group such as a family." Examples of sentiments include romantic love, parental love, loyalty, friendship, patriotism, hate, as well as more transient, acute emotional responses, to social losses (sorrow, envy) and gains (pride, gratitude) [2].

B. *Overview of Opinion:*

In the case of opinions, not all words used in the sentence have significance. Some words are classified as noise because they are of no use in the process of classifying the polarity of the opinion.

According to Kim and Hovy [4], an opinion consists of the following four parts: topic, opinion holder, claim, and sentiment. That is, for each opinion, there is a holder who believes a claim about a topic and then associates a positive, negative, or neutral (neutral here does not mean absence, e.g., "The winter has arrived." It is not good or bad, just saying sentiment with the claim).

Fig. 1. Example of an Opinion(based on Kim and Hovy[4])

Fig.1 illustrates how the four aspects can appear in a sentence. Kim and Hovy [4] further pointed out that an opinion can be subjective without implying a sentiment.

This paper is made further as: Section II discusses related work examined till now. Section III describes overall system design. Section IV presents usage points of interest, calculation utilized and scientific model. Section V closes with the conclusions and presents future work.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

II. RELATED WORK

With the rapid growth of online reviews, review mining has attracted a great deal of attention. Early work in this area was primarily focused on determining the semantic orientation of reviews. Whitelaw et al. [5] defined the concept of "adjectival appraisal groups" headed by an appraising adjective and optionally modified by words like "not" or "very." Each appraisal group was further assigned four types of features: orientation, polarity, graduation and attitude. They reported good classification accuracy using the appraisal groups. They had also shown that the classification accuracy can be further increased when they are combined with standard "bag-of-words" features.Chien-Liang Liu, Wen-Hoar Hsaio, Chia-Hoang Lee, Gen-Chi Lu, and Emery Jou done movie rating and review summarization in mobile environment [6].Theysuggested a novel approach based on latent semantic analysis (LSA) to identify product features. Furthermore, they come to light a way to reduce the size of summary based on the service/product features obtained from latent semantic analysis (LSA). They took into consideration both sentiment-classification accuracy and system response time to design the system. Kamps and Marks used "Words with Attitude" approach [7]. In that, they suggested a system to calculate the semantic distance from a word to good/bad. There also studied that work at a finer level and used strings/words as classification subject. They categorize words into two groups, "good" and "bad" and then use certain functions to evaluate the overall "goodness" or "badness" score for the documents. Some authors had used the approach to record different rating scales to recognize opinions of users. They used the stars rating aspects and recorded online reviews of users on different services/products. These recorded stars rating is then used to recognize the service/product performance [8]. In most of studies mentioned above, the sentiments are captured by explicit rating scales such as the number of stars; few studies have attempted to enhance text mining strategies for sentiment categorization. Ghose and Ipeirotis [9] argued to fill in this gap, that review texts contain wealth of information that cannot be easily captured using simple numerical ratings. In their study, they assigned a "dollar value" to a collection of adjective-noun pairs, as well as adverb-verb pairs, and investigated how they affected the bidding prices of various products at Amazon. Xin Chen, Mihaelavorvoreanu and Krishna madhavan [1] used multi-label classifier for mining engineering student's problem but they only considered negative aspects of engineering student's problem. M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas proposed SentiStrength to detect the strength of sentiment in short, informal conversation of public on social media, with a focus on MySpace comments. The algorithm provides two ordinal scales of valence for positivity and negativity ({+1… +5} and {-5…-1}, respectively) [10]. It utilizes an extended version of the affective dictionary from the "Linguistic Inquiry and Word Count" (LIWC) software, which contains a list of positive and negative emotional bearing words, each one annotated with a value of 1 to 5, indicating its sentiment strength [11].

III. IMPLEMENTATION DETAILS

A. *System Outline:*

Fig. 2depicts the overall system architecture presented in this paper for classifying the tweets as well as analysing the sentiment for respective tweet. The main methodology behind sentiment analysis is the classifier method specifically multinomial/multiclass naïve bayes classifier where twitter data i.e. tweet is being classified as positive, negative or other which in turn classified as happy, sad, angry or disgusting in subcategory under the above given category. The framework shown above consist four modules viz. data collection, data pre-processing, machine training & testing for sentiment classification and web portal. The front end of above mentioned framework is a web portal with one search box. The user will enter the query relevant to political domain. Tweets containing that searched keyword will be fetched from data source and data source is populated by twitter streaming API. The input given to sentiment analysis engine is the texts of all those collected tweets where it produces each tweet with particular category that is with sentiments of tweet to identify public's positive as well as negative aspects.

First of all, we have collected tweets from the twitter using search API which is open source. It will dump the old tweets into the data source; with the use of these old tweets we have trained the classifier with machine learning algorithm, so that it can work finely with unknown tweets. The new tweets will always be populated; and final category of that tweet is predicted by classifier. The machine is trained with classifier also so that it can classify the retrieved tweets into particular category.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

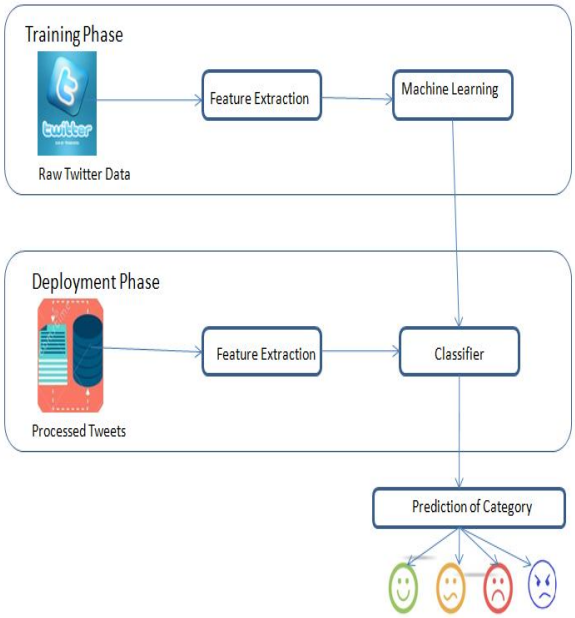

Fig. 2. System Architecture

- Data Collector

We have collected tweets from twitter as our input to trained sentiment classification engine. Twitter provides two types of APIs to dump tweets: Search API used for dumping old tweets while Streaming API used for dumping live tweets. Using Search API, we have built training data set for sentiment classification engine and by using streaming API; it will display current results too.

- Data Preprocessing:

Dumped tweets from twitter always contain the URL's, usernames, hashtags which makes no sense for sentiment classification & hence removed, which involve following subtasks:

1. Removing User Names:

In tweets user may refer to another user with '@' symbol. In data classification these user names never play any important role so they are removed.

2. Removing URL's:

In some tweets user post some hyperlinks related to that tweet. In data classification these URL's never play any role. With the help of regular expression we need to remove URL's from tweets.

3. Removing Hashtags(#):

People use the hashtags symbol # before a relevant keyword or phrase (no spaces) in their tweet to categorize those tweets and help them show more easily in twitter Search. In data classification these hashtags never play any important role so they are removed with the help of regular expression.

Once the Data preprocessing step is over, these processed tweets are stored in Data source.

- Model Training & Testing:

We have implemented multi-class text based classifier like Bayesian classifier. In it we have built a probabilistic classifier based on modeling the underlying word features in different classes. Then we have classified text based on posterior probability of the documents belonging to the different classes on the basis of word presence in the document. Once the classifier is ready in sentiment classification engine, we have trained the machine by using training set built in

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

previous step. This trained model is then tested for accuracy. Whatever the tweets that we have stored in data source after first and second module, out of which 80% are used as trained data and 20% is used for testing data.

- Deployment of Web Portal

In this module we have developed one portal with one search box where user will enter query related to political domain. Tweets containing above searched query are then collected from data source which is populated by Streaming API of twitter. Text of all these retrieved tweets is then passed to trained sentiment classification engine. Each tweet then returned with particular category. User will get back the result or tweets containing the searched keyword in a proper graphical format along with their classification.

B. *Data Set:*

Fig. 3. Number of tweets in each category for training data.

From the above Fig 3, it is clear that we have collected total 350 tweets for training machine.

C. *Pseudo Code:*

Step 1: Let, D: = {set of documents to classify};
Step 2: D :={d_1,d_2,....,d_n};
Step 3: W: = {set of categories};
Step 4: W :={ w_1, w_2,....,w_n};
Step 5: X: ={ set of tokens in particular document};
Step 6: P: = {set of probabilities for each class for respective document};
Step 7: for each di in D
X: ={ set of tokens from di};
for each w_j in W
product := 1;
for each x_k in X

$$P(x_k| w_j) = \frac{\Sigma\, tf(xk,di\epsilon\, wj) + \alpha}{\Sigma\, Nd\, \epsilon\, wj + \alpha.V}; \quad \text{eq. (1)}$$

product: = product * p (x_k/w_j);
end for
add product to P;
end for
Max(P);
Step 8: end for.

Where,

- $\Sigma\, tf(x_i , d \in \omega_j)$: The sum of raw term frequencies of word xi from all documents in the training sample that belong to class ω_j .
- $\Sigma\, Nd \in \omega_j$: The sum of all term frequencies in the training dataset for class ω_j .
- α: An additive smoothing parameter (α = 1 for Laplace smoothing).
- V: The size of the vocabulary (number of different words in the training set).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

D. *Basic Naïve Bayes Classifier:*

Naïve Bayes classifier is known for classifying the text documents based upon the Bayesian methodology which represents a statistical method as well as a supervised learning method for classification. It is particularly used when dimensionality of the inputs is high. Naïve Bayes classifier work on the principle of calculating the conditional probabilities whether the document is positive, negative or in other category. Calculating conditional probability is a method of estimating the maximum likelihoods of term within the document.

Now there are several Naive Bayes Variations, out of which we are interested in multinomial naïve Bayes that is designed more for text documents. Simple naive Bayes would model a document as the presence and absence of particular words; multinomial naive Bayes explicitly models the word counts and adjusts the underlying calculations to deal with in [12].

E. *Tuning Multinomial Naïve Bayes:*

Important factor while training Naive Bayes classifier for text classification is to calculate term frequency (tf(t, d)). These term frequencies are then used to calculate likelihood of that term in particular class as follows [13]

$$\hat{P}(x_i \mid \omega_j) = \frac{\sum tf(x_i, d \in \omega_j) + \alpha}{\sum N_{d \in \omega j} + \alpha \cdot V}$$

eq. (2)

The class conditional probability for that document is then calculated as the product from the likelihoods of the individual words as follows

$$P(\mathbf{x} \mid \omega_j) = P(x_1 \mid \omega_j) \cdot P(x_2 \mid \omega_j) \cdot \ldots \cdot P(x_n \mid \omega_j) = \prod_{i=1}^{m} P(x_i \mid \omega_j)$$

q. (3)

All terms are considered equally important for determining relevancy. Now certain terms have little or no discriminating power in determining relevance. For example suppose we have collection of documents for automobile industry in which all document are likely to have term auto. By above formula this term is adding its effect in each document while determining its relevancy i.e. category. So we have introduced a mechanism for attenuating the effect of terms that occur too often in the documents to be meaningful for relevance determination. The idea behind this is to scale down the term weights for terms with high document frequency.

So we have followed Tf-idf i.e. Term Frequency - Inverse Document Frequency. The Tf-idf approach assumes that the importance of a word is inversely proportional to how often it occurs across all documents [14]. Now Tf-idf for each term can be calculated as,

Tf-idf = tfn(t,d)*idf(t);eq. (4)

Where,

idf(t) = log (Total number of documents / Number of documents with term t in it)

F. *Mathematical Model :*

$S=\{I, O, F_{PROBABLISTIC}, T_C\}$

$I = \{I_1, I_2 \ldots I_N\}$

I =is a set of inputs where input is given as a tweet whose sentiment is to be determined.

$O = \{PL_1, PL_2, \ldots, PL_N\}$

where n is number of categories.

O= set of probability of each class for that tweet.

$F_{PROBABLISTIC}$= Function to calculate probability of tweet for each category

$F_{PROBABLISTIC}$ $\quad MAX\ ()$

$I \quad O \quad \rightarrow \quad T_C \quad \rightarrow$

T_C= Final category for respective tweet.

Success = $T_C \neq \Phi$

Failure = $T_C = \Phi$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

IV. EXPERIMENTAL RESULT & ANALYSIS

As mentioned earlier we have used 350 tweets for training machine. Training machine means calculating no. of unique words in whole corpus, term frequency of each term in all documents for each category. We have trained machine in both ways, we will call them as classifier-1 and classfier-2. For classifier-2 we have used apache mahout as framework for training classification machine which follows TF (t) * IDF (t) approach [15],

i. Calculating only term frequencies (classifier-1)

TF (t) = (Number of times term t appears in a document) / (Total number of terms in the document).

ii. Multiplying term frequencies with inverse document frequency (classifier-2)

IDF (t) = **log** (Total number of documents /Number of documents with term t in it)

After training these two machines we have used 50 unknown tweets from same domain i.e. politics as test data. After running two trained machines on this test data we came with following results.

Table 1.Performance Comparison

Classifier/results	No of correctly classified tweets	No of incorrectly classified tweets
classifier-1	31	19
classifier-2	39	11

Table 1 gives experimental results. From these results it is clear that classifier-1 comes with 62% accuracy while classifier with term frequency multiplied by inverse document frequency i.e. classifier-2 comes with 78% accuracy.

A. *Complexity Analysis :*

Time complexity for classifying tweet can be determined as follows:

```
Step 1: Map<category, product> map;
Step 2: Foreach category
        product:=1;
        foreach token in tweets
        calculate p(xi/wj);
                product := product * p(xi/wj);
        end;
        map.put(category, product);
Step 3: End;
Step 4: FinalCategory:=map.getKey(Max(map.get(product)));
```

So for each category, product will be calculated for each token in tweets; suppose there are N categories and M tokens in tweets then complexity will be

$$O(N*M); \text{eq. (5)}$$

V. CONCLUSION AND FUTURE WORK

Our study is advantageous to researchers in political data mining domain, learning technologies and learning analytics. It provides a workflow for analyzing social issues, social media data on political domain which overcomes the major drawback of both manual qualitative analysis and large scale computational analysis of user generated textual content. Our study can inform political administrators, practitioners, thinkers and other relevant decision makers to gain further understanding of overall scenario of politics within that region, state, and country.The system can be extended to do sentiment analysis on any domain like movie, songs, and product feedbacks just by changing training corpus.The system can be extended to work with Marathi or Hindi language data.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 6, June 2015

REFERENCES

1. X.Chen, M.Vorvoreanu, and K.Madhavan, "Mining Social Media Data for Understanding Students' Learning Experiences," IEEE transactions on learning technologies vol. 7, no. 3, July-September 2014.
2. MyriamMunezero, CalkinSuero Montero, ErkkiSutinen, and John Pajunen, "Are They Different? Affect, Feeling, Emotion, Sentiment, and Opinion Detection in Text" in IEEE transactions on affective computing, vol. 5, no. 2, april-june 2014.
3. S. L. Gordon, "The sociology of sentiments and emotion," in Social Psychology: Sociological Perspectives, M. Rosenberg and R. H. Turner, eds., New York, NY, USA: Basic Books, 1981 pp. 562–592.
4. S. Kim and E. Hovy, "Determining the sentiment of opinions," in Proc. 20th Int. Conf. Comput.l Linguistics, PA, USA, 2004, pp. 1367–1363.
5. C. Whitelaw, N. Garg, and S. Argamon, "Using Appraisal Groups for Sentiment Analysis," Proc. 14th ACM Int'l Conf. Information and Knowledge Management (CIKM), pp. 625-631, 2005.
6. Chien-Liang Liu, Wen-Hoar Hsaio, Chia-Hoang Lee, Gen-Chi Lu, and Emery Jou, "Movie Rating and Review Summarization in Mobile Environment" in IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 42, no. 3, may 2013
7. Kamps and M. Marx, "Words with Attitude," Proc. First Int'l Conf. Global WordNet, pp. 332-341, 2002.
8. B. Pang and L. Lee, "Seeing Stars: Exploiting Class Relationshipsfor Sentiment Categorization with Respect to Rating Scales," Proc.43rd Ann. Meeting on Assoc. for Computational Linguistics (ACL),pp. 115-124, 2005.
9. A. Ghose and P.G. Ipeirotis, "Designing Novel Review Ranking Systems: Predicting the Usefulness and Impact of Reviews," Proc. Ninth Int'l Conf. Electronic Commerce (ICEC), pp. 303-310, 2007.
10. M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment Strength Detection in Short Informal Text," J. Am. Soc. Information Science and Technology, vol. 61, pp. 2544-2558, Dec. 2010.
11. M. Francis, J. Pennebaker, and R. Booth, Linguistic Inquiry and Word Count: LIWC, second ed. Erlbaum Publishers, 2001.
12. "A Comparison of Event Models for Naive Bayes Text Classification" byAndrew McCallum and Kamal Nigam
13. http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html
14. http://www.tfidf.com/
15. https://mahout.apache.org/users/basics/quickstart.html

BIOGRAPHY

Amruta U. Tarlekarhas received the Bachelor'sdegree in Computer Science from BharatiVidyapeeth College of Engineering, Kolhapur, India in 2012 and perusing masters in Computer Engineering from TSSM's BhivrabaiSawant College of Engineering, Pune, India. She is currently working as Assistant Professor with G.S.Moze College of Engineering, Pune. Her research interest is data mining & learning analytics.

Prof. M. K. Kodmelwaris a full time Assistant Professor at Department of Computer, TSSM's BSCOER,Narhe,Pune,India.He has 14 years of experience in teaching. He is currently perusing Ph.D in Computer Science and completed his Master's degree from BharatiVidyapeeth, Pune and his research interest is computer network.