Volume 7, Issue 4, April 2017 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Controversial Analysis:- Sentimental Analysis of Twitter Data

Samarth Jaykar Shetty, Badal Rakesh Thosani, Lenherd Deon Olivera, Supriya Kamoji

Department of Computer Science Engineering, Fr. Conceicao Rodrigues College of Engineering, Mumbai, Maharashtra, India

Abstract: Controversial analysis deals with identifying and classifying opinions or sentiments expressed in source text. We present a novel approach for naturally ordering the sentiments of Twitter messages. These messages are delegated positive or negative or neutral concerning a query term. This is valuable for buyers who need to examine the notion of items before buy, or organizations that need to screen the general public sentiment of their brands. Past research on characterizing opinion of messages on microblogging administrations like Twitter have attempted to tackle this issue however have disregarded neutral tweets which prompts to wrong feeling characterization and we have attempted to take care of this issue in this project. We show the consequences of machine learning algorithm for classifying the sentiment of Twitter messages utilizing a novel feature vector. Our training data comprises of openly accessible twitter messages acquired through mechanized means. We demonstrate that machine learning algorithm (Naive Bayes and SVM) can accomplish aggressive exactness when prepared utilizing our feature vector and the freely accessible dataset. This report also describes the pre-processing steps of the dataset required to accomplish high accuracy. The primary commitment of this project is the novel feature vector of weighted unigrams used to prepare the machine learning classifiers.

Keywords: Twitter, Sentiment Classification, Sentiment Analysis.

I. INTRODUCTION

Twitter is a famous microblogging administration where clients make status messages (called "tweets"). These tweets some of the time express sentiments about various subjects. We propose a strategy to naturally remove slang (positive or negative or neutral) from a tweet. This is exceptionally valuable since it permits criticism to be collected without manual intercession. Buyers can utilize assumption examination to research items or administrations before making a buy. Advertisers can utilize this to research popular supposition of their company and product, or to analyse customer satisfaction. Associations can likewise utilize this to assemble basic criticism about issues in recently released products. There has been a lot of research in the territory of sentiment classification. Generally the majority of it has concentrated on ordering bigger bits of content, similar to surveys [9]. Tweets (and microblogs in general) are different from reviews primarily because of their purpose: while reviews represents to condensed contemplations of author, tweets are more casual and constrained to 140 characters of content. For the most part, tweets are not as insightfully formed as reviews. However, regardless they offer organizations an extra road to accumulate criticism. Past research on examining blog posts incorporates [6]. Pang et al [9] have break down the execution of various classifiers on movie review. The work of Pang et al. has filled in as a gauge and many authors have utilized the systems in their paper across various domain. In order to train a classifier, supervised learning usually requires hand-labelled training data.

With the substantial scope of points talked about on Twitter, it would be exceptionally hard to physically gather enough information to train a sentiment classifier for tweets. For positive, negative and neutral tweets, we have utilized the openly accessible tweet dataset.[13]. We have run the machine learning classifiers Naïve Bayes and Support Vector Machine trained on the positive and negative tweets dataset and the neutral tweets against a test set of tweets.

1.1 Defining the sentiment

With the end goal of research, we have characterized supposition to be "an individual positive or negative feeling" and when there is a non appearance of this, we regard it as a neutral conclusion. Table 1 shows some examples.

Table 1. Example Tweets

Sentiment	Keyword	Tweet
Positive	Cricket	Dammmmm I Love Cricket
Negative	Gary Kirsten	Gary Kirsten to resign as Indian Coach

Neutral	airplane	Come 10 a clock, phone going on airplane mode

1.2 Characteristic of Tweets

Twitter messages have numerous exceptional traits, which separates our work from past research:

A. Length

The most extreme length of a Twitter message is 140 characters. This is very different from the previous sentiment classification research that focused on classifying longer bodies of work, such as movie reviews.

B. Language model

Twitter clients post messages from various media, including their mobile phones. The recurrence of incorrect spellings and slang in tweets is significantly higher than in different areas.

C. Domain

Twitter clients post short messages about a variety of topics unlike other sites which are tailored to a specific topic. This contrasts from a substantial rate of past research, which concentrated on particular domain, for example, movie review.

II. RELATED WORK

This project expands on the thoughts proposed [12] where the authors order tweets utilizing unigram feature and the classifiers are trained on data obtained using distant supervision.The [10] demonstrates that utilizing emoticons (distant supervision) as marks for positive and assessment is effective for reducing dependencies in machine learning methods and this thought is intensely utilized[12].Pang and Lee [9] explored the execution of different machine learning methods in the particular domain of motion picture surveys.

However, the past strategies have not considered neutral tweets which prompt to wrong characterization and this project tries to tackle this issue by incorporating neutral tweets in the training dataset and utilizing a novel feature vector to prepare the machine learning classifier and tries to order a given tweet as positive or negative or neutral.

III. APPROACH

Our approach is to utilize diverse machine learning classifiers and feature extractors. The machine learning classifiers are Naive Bayes and Support Vector Machines (SVM). The component extractors are unigrams and unigrams with weighted positive and negative keywords .We assemble a structure that regards classifiers and feature extractors as two distinct parts. This structure permits us to effectively experiment with various blends of classifiers and feature extractors.

3.1 Query Term

We standardize the impact of query terms. Our suspicion is that the client needs to perform notion investigation about a product and not of a product. In the event that a query term has a positive/negative feeling without anyone else's input, this will bias the outcomes.

3.2 Emotion

Use of emoticons is very prevalent throughout the web, more so on micro- blogging sites. We identify the following emoticons and replace them with a single word. Table 2 lists the emoticons we are currently detecting. All other emoticons would be ignored.

Table 2. List of Emoticons

Emotions	Examples					
EMOT_SMILEY	:-)	:)	(:	(-:		
EMOT_LAUGH	:-D	:D	X-D	XD	xD	
EMOT_LOVE	<3	:*				
EMOT_WINK	;-)	;)	;-D	;D	(;	(-;
EMOT_FROWN	:-(	:(	(:	(-:		
EMOT_CRY	:,(	:’(	:”(	:((		

3.3 Feature reduction

The Twitter language model has many unique properties. We take advantage of the following properties to reduce the feature space.

A. Username

Clients regularly incorporate Twitter usernames in their tweets keeping in order to direct their messages. An accepted standard is to incorporate the @ symbol before the username (e.g. @Badal). An equivalence class token (AT_USER) replaces all words that begin with the @ symbol.

B. Usage of links

Clients all the time incorporate links in their tweets. A proportionality class is utilized for all URLs. That is, we change over a URL like "https://plus.google.com/111998632899877039974" to the token "URL".

C. Stop words

There are lots of stop words or filler words, for example, "a‟, "is", "the" utilized as a part of a tweet which does not show any sentiment and subsequently these are filtered out. The entire rundown of stop words can be found at [14].

D. Repeated letters

Tweets contain extremely casual language. For instance, if you search "hungry" with a discretionary number of u’s in the centre (e.g. huuuungry, huuuuuuungry, huuuuuuuuuungry) on Twitter, there will doubtlessly be a non empty result set. We utilize pre-processing so that any letter happening more than two circumstances in succession is supplanted with two events. In the examples over, these words would be changed over into the token "huungry".

Table 3. Effect of Feature Reduction

Feature Reduction Steps	**# of Features**	**Percentage of Original**
None	9,77,354	100%
URL/Username/Repeated letters	3,60,644	36.90%
Stop Words(‘a’,’is’,’the’)	85,421	8.74%
Final	85,421	8.74%

3.4 Feature vector

In the wake of pre-processing the training set information which comprises of 35,000 positive tweets, 35,000 negative tweets and 20,000 neutral tweets, we figure the feature vector as underneath:

A. Unigrams

As appeared in Table 3, toward the finish of pre-processing we wind up with 85,421 elements which are unigrams and each of the components have equal weights.

B. Weighted Unigrams

Rather than measuring each of the unigrams similarly, we present bias by measuring the positive and negative keywords more than alternate elements introduced in the feature vector. We utilize our point of convergence (a rundown) so as to weight the positive and negative keywords more contrasted than the rest of the elements[15][16].

IV. MACHINE LEARNING METHODS

We tested different classifiers namely Naïve Bayes and Support Vector Machines.

4.1 Naïve Bayes

Naive Bayes is a simple model which works well on text categorization [5]. use a multinomial Naive Bayes model. Class c ∗ is assigned to tweet d, where

$$c* = argmac_c P_{NB}(c|d)$$

$$P_{NB}(c|d) := \frac{(P(c) \sum_{i=1}^{m} P(f|c)^{n_i(d)})}{P(d)}$$

In this formula, f represents a feature and ni(d) count of feature fi found in tweet d. There are a total of m features. Parameters P(c) and P(f|c) are obtained through maximum likelihood estimates, and add-1 smoothing is utilized for unseen features. We used the Python based Natural Language Toolkit [17] library to train and classify using the Naïve Bayes method.

4.2 Support Vector Machines

Support Vector Machines is another popular classification technique [2]. We have used libsvm [18] library with a linear kernel. Our input data are two sets of vectors of size m. Each entry in the vector corresponds to the presence a feature. In the unigram feature extractor, each feature is a single word found in a tweet. If the feature is present, the value is 1, but if the feature is absent, then the value is 0.We use feature presence, as opposed to a count, so that We do not have to scale the input data, which speeds up overall processing [1].

V. EVALUATION

5.1 Experimental setup

There are publicly available data sets of twitter messages with sentiment indicated by [12] and [13]. We have used a combination of these different datasets to train the machine learning classifiers. For the test dataset, We randomly choose tweets which were not used to train the classifier.

Table 4. Example Tweets

Dataset	Positive	Negative	Neutral	Total
Train Data	35,000	35,000	20,000	90,000
TestData	Randomly chosen Tweets			15,000

The Twitter API has a parameter that specifies in which language to retrieve the tweets .We always set this parameter to English (en). Thus, our classification will only work on tweets in English because the training data is English-only.

VI. RESULT

We explore the usage of unigrams and weighted unigram feature and Table 5 summarizes the results.

Table 5. Classifier Accuracy

Features	**Naïve Bayes**	**SVM**
Unigrams	69.82%	75.90%
Weighted Unigrams	81.52%	92.42%

A. Unigrams

The unigram feature vector is the simplest way to retrieve features from a tweet. The machine learning algorithms perform average with this feature vector. The accuracy obtained by the unigram approach is lower as compared to the weighted unigram approach.

B. Weighted Unigrams

In this approach, We took advantage of the fact that it makes sense to weight the positive and negative keywords more than other words while trying to classify the sentiment of a tweet and this trick produced competitive accuracy as shown in Table 5 . As expected, SVM performed the best with 92.42% accuracy and it performed better than Naïve Bayes which produced an output of 81.52%.

VII. FUTURE WORK

Machine learning techniques perform well to classify slant in tweets. We trust the exactness of the framework could be as yet progressed. The following is a rundown of thoughts we think could help the arrangement:-

A. Semantics

The algorithms characterize the overall sentiments of a tweet. The extremity of a tweet may rely on upon the viewpoint you are translating the tweet from. For instance, in the tweet "India beats Japan :)", the feeling is sure for India and negative for Japan. For this situation, semantics may offer assistance. Utilizing a semantic part labeller may demonstrate which thing is for the most part connected with the verb and the grouping would happen in like manner. This may permit "Japan beats India:)" to be grouped uniquely in contrast to "India beats Japan- :)".

B. Bigger dataset

The training dataset in the request of millions will cover a superior scope of twitter words and thus better unigram feature vector bringing about a general enhanced model. This would immeasurably enhance the current classifier results

C. Internationalization

As of now, we concentrate just on English tweets however Twitter has a colossal universal group of onlookers. It ought to be conceivable to utilize our way to deal with sentiment in different languages with a language specific positive/negative keyword list.

D. User Interface

To help visualize the utility of the Twitter-based sentiment analysis tool, we will fabricate a web application tool.This can be utilized by people and organizations that might need to research sentiment on any subject. We will build a web interface which searches the Twitter API for a given keyword for the past one day or seven days and fetches those results which is then subjected to preprocessing. These filtered tweets are fed into the trained classifiers and the resulting output is then shown as a graph in the web interface.

VIII. CONCLUSION

Machine Learning techniques can be applied for twitter sentiment analysis. There are certain issues while dealing with identifying emotional keyword from tweets having multiple keywords. It is also difficult to handle misspellings and slang words. To deal with these issues, an efficient feature vector is created by doing feature extraction in two steps after proper preprocessing. In the first step, twitter specific features are extracted and added to the feature vector. After that, these features are removed from tweets and again feature extraction is done as if it is done on normal text. These features are also added to the feature vector. Classification accuracy of the feature vector is tested using different classifiers like Naive Bayes, SVM. Utilizing a novel feature vector of weighted unigrams, we have demonstrated that machine learning algorithms, for example, Naïve Bayes and Support Vector Machines accomplish aggressive exactness in characterizing tweet sentiments.

REFERENCES

[1] D. O. Computer, C. wei Hsu, C. chung Chang, and C. jen Lin. A practical guide to support vector classification chihwei hsu, chih-chung chang, and chih-jen lin. Technical report, 2003.

[2] N. Cristianini and J. Shawe-Taylor. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, March 2000.

[3] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Microblogging as online word of mouth branding. In CHI EA "09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, pages 3859–3864, New York, NY, USA, 2009. ACM.

[4] T. Joachims. Making large-scale support vector machine learning practical. In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in kernel methods: support vector learning, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.

[5] C. D. Manning and H. Schutze. Foundations of statistical natural language processing. MIT Press, 1999.

[6] G. Mishne. Experiments with mood classification in blog posts. In 1st Workshop on Stylistic Analysis Of Text For Information Access, 2005.

[7] K. Nigam, J. Lafferty, and A. Mccallum. Using maximum entropy for text classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61–67, 1999.

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

[9] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79–86, 2002.

[10] J. Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In Proceedings of ACL-05, 43nd Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2005.

[11] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), Vancouver, CA, 2005.

[12] Alec Go, Richa Bhayani, Lei Huang. Twitter Sentiment Classification using Distant Supervision. Technical report, Stanford Digital Library Technologies Project, 2009.

[13] Publicly available dataset :- https://github.com/badalthosani/Controversial-Analysis/blob/master/final.csv.xlsx

[14] Stop words list – https://github.com/badalthosani/Controversial-Analysis/blob/master/Stopwords.txt

[15] Positive keyword list - https://github.com/badalthosani/Controversial-Analysis/blob/master/pos_mod.txt

[16] Negative keyword list – https://github.com/badalthosani/Controversial-Analysis/blob/master/neg_mod.txt

[17] Python Natural Language Toolkit - http://www.nltk.org/

[18] Libsvm:-http://www.csie.ntu.edu.tw/~cjlin/libsvm/