SENTIMENT ANALYSIS OF TWITTER DATA SET

Lakshmishree.C[1], Niharika.W.M[2], Neha.P[3]

Student, Computer Science and Engineering, M.S.Ramaiah Institute of Technology, Bangalore, India [1, 2, 3]

Abstract: This paper underscores an approach for the automatic classification of the sentiment of Twitter messages. The tweets are categorized as positive, neutral or negative with respect to a query term. This can be used by consumers who want to determine the sentiment held by the general public towards a particular product before they make a purchase, or companies who want to monitor the public sentiment of their brands and products. The classification of tweets is done by employing the Naïve Bayes Algorithm. Since this is a supervised machine learning algorithm, a publicly available dataset of twitter messages is used to train the Naïve Bayes classifier. The preprocessing of the dataset done to achieve high accuracy is also described. The primary focus of this project is the usage of a novel feature vector of unigrams to train the machine learning classifier and test it.

Keywords: Twitter, Naïve Bayes Classifier, Unigrams, Feature Vector.

I. Introduction

Twitter is a well-known microblogging service where users post short, 140 character status messages called as “tweets”. The general public sometimes utilizes the Twitter platform to express their opinion regarding various products, services, etc. The opinions expressed by people through their tweets are greatly valuable as they reflect the impromptu or unrestrained thoughts of people in contrast to the cautiously engineered product or service reviews. This paper describes the building of an automated sentiment analyser of tweets. The ability of the sentiment analyser to aggregate the public opinion in a large set of tweets without any manual intervention makes this approach very desirable. This sentiment analyser can find its application in several areas – for consumers, who want to examine the reviews of the product they plan to purchase, marketers can determine the general opinion held by the public regarding their product or service, and organizations will be able to gather critical feedback of their newly released products.

The machine learning algorithm used for sentiment analysis of tweets is the Naïve Bayes Classifier. This algorithm is supervised, which implies that it has to be manually trained before being used to classify tweets. Also, we use the unigrams approach for sentiment analysis of tweets. In the unigrams approach, each word of the tweet is considered individually, independent of the other words to determine its sentiment.

Specifically, this paper aims to extract tweets relating to the hashtags of various smartphones and use the sentiment analyzer built to determine the most popular smartphone. To help visualize the utility of the Twitter-based sentiment analysis tool, we have used SAP Lumira for visualization.

A. *Tweets and their Sentiment*

As stated earlier, tweets are succinct, 140 character text messages. A sentiment can be simple defined as a personal view or opinion that is held. It may be positive or negative, and when there is a lack of this personal opinion, the sentiment is categorized as neutral. Sentiment analysis, also known as opinion mining aims to identify and extract subjective information from source materials which can be positive, neutral, or negative [1].

B. *Characteristics of Tweets*

It is necessary to understand the unique attributes possessed by Tweets. First, length, it represents the maximum length of a Twitter message of 140 characters. Second, the language model of Twitter is such that users post messages from many different media, including their cell phones, tablets, PC’s, and laptops. There is a high tendency of use of informal language, several words may be misspelt, and several word are replaced by their shorter forms. Third, tweets have a very vast domain.

II. Design

A. *Approach*

Our approach is to use Naïve Bayes machine learning classifier and feature extractor. The feature extractors we have used are unigrams [1].

1) Query Term

Our assumption is that the user has a specific product over which he wants to extract related tweets and perform sentiment analysis. The results are certainly biased if the query positive or negative [1].

2) Emoticons

During the training process itself, the emoticons are categorized as noisy labels and hence all tweets used in training are stripped of emoticons. Consequently, even during classification of tweets the tweets are stripped of emoticons. The classifier is not trained to process emoticons present in the tweet. The classifier uses these non-emoticon features to determine the sentiment [1].

3) Feature Reduction

The Twitter language model has several unique properties. We take advantage of the following properties to prevent the feature vector from exploding – usernames, URL’s, stop words, and repeated letters. All these are explained in detail in next section [1].

4) Feature vector:

Next, we need to compute the feature vector. The presence and absence of words are the features of tweets.

During the training of the classifier, each of the positive, negative and neutral tweets can be split into individual words. Each of these words are appended to the feature vector. The words which will not contribute to the determining of the sentiment are filtered out. The process of adding individual (single) words to the feature vector is referred to as 'unigrams' approach.

The sentiment analyser is designed to perform the following steps before the actual determination of the sentiment of the tweet – training of the Naïve Bayes classifier, pre-processing of input tweets, filtering of tweets, formation of the feature vector, and finally testing the trained classifier with the tweets as input [1].

III. Implementation

A. *Extraction of tweets*

In order to classify the sentiment of tweets, first, a database of tweets has to be established. About 1 lakh tweets are extracted from the Twitter streaming API from the hashtags relating to different smartphones and stored in a SQLite database. Twitter API places a limit of 150 unauthorized requests per hour and hence one can download only 3600 tweets per day via the labelled tweet IDs from the publicly available tweet ID dataset. The retrieved data is subjected to preprocessing and is used as a test data for analysing sentiments using classification algorithm.

Consider a set of sample positive, negative and neutral tweets.

A few sample positive tweets are:

- @PrincessSuperC hey cici sweetheart! just wanted to let u know i luv my new iPhone!
- @Msderbraymme i heard about your new Nexus! congrats girl!!
- New Nexus on 5th!! Its Awesome!!.Check it out on : www.nexus.com

A set of sample neutral tweets are:

- Apple iPhone 6s and 6s Plus: Australian pricing and availability http://t.co/DLtDw3rNAG #iphone #iphone6s
- Everything You Need To Know About #Nexus http://t.co/OenjEqQLXB #AppleEvent #Smartphones
- The only thing that's changed is #iPhone6s #Apple

A set of sample negative tweets are:

- The New iPhone Is The Heaviest iPhone Ever Made http://t.co/mCrupweXOD via @dadaviz #iphone #iphone6s #iphone6plus @apple
- Smashed my iPhone 6. Must be a sign #iphone6s #ShotOniPhone6
- @charles #nexus How expensive is the new Nexus??

B. *Preprocessing*

As seen above, the tweets contain some valuable information which contribute to the deriving of the sentiment and a set of words which do not contribute to the sentiment. Hence, we find the need to preprocess them [1]. This stage involves performing of the following actions [1]:

1). Conversion to lower case

All characters of the tweet are converted to lower case letters for uniformity.

2). Elimination of URL's

All URL's are gotten rid of using regular expressions. Sometimes, a URL in a tweet can also be replaced by a generic word URL. We do this because following the page of the URL and determining its sentiment is very complex.

3). Elimination of usernames

Users often include Twitter usernames in their tweets. A de facto standard is to include the @ symbol before the username. Such usernames can be eliminated using regular expression matching. Alternatively, they can be replaced by the generic term AT_USER.

4). Processing of hashtags

Hashtags can be useful to determine the specific topic over which a tweet conveys an opinion. Hence, the hashtag is replaced with the word without the hash symbol. Ex. #iphone6 is replaced with iphone6.

5). Elimination of punctuation and additional whitespaces

Punctuations at the start and the end of the tweet are removed. It is also helpful to replace multiple white spaces by a single space.

After preprocessing, the sample tweets look as follows:

The positive tweets after preprocessing:

- AT_USER hey cici sweetheart! just wanted to let u know i luv my new iphone
- AT_USER i heard about your new nexus! congrats girl
- new nexus on !! its awesome!!.check it out on URL

The neutral tweets after preprocessing:

- apple iphone and plus: australian pricing and availability URL iphone iphone6s
- everything you need to know about nexus URL appleevent smartphones
- the only thing that's changed is iphone6s apple

The negative tweets after preprocessing:

- the new iphone is the heaviest iphone ever made URL via AT_USER iphone iphone6s iphone6plus AT_USER
- smashed my iphone. must be a sign iphone6s shotoniphone6
- AT_USER nexus how expensive is the new nexus

C. *Feature Engineering*

Feature Extraction is an elemental task for Sentiment Analysis. Converting a piece of text to a feature vector is the basic step in Sentiment Analysis. After the tweets have been subjected to the previous phases, a feature vector of the tweet results. Feature Vector is the most important concept in building a sentiment analyser. The success of the classifier is directly dependent on the construction of a good feature vector. The feature vector is used to build a model where the classifier learns from the training data and then functions to classify new, unseen data. In tweets, the presence and absence of words that appear in a tweet as features. For text classification purpose, the unigram model was used which selects individual words from the data. [3].

Filtering is an essential sub-task of building of a feature vector. Filtering of tweets involves performing of the following tasks:

1). Removal of stopwords

Words such as – a, as, with, that, the, etc., are called stopwords. They do not contribute to the determining of a sentiment and hence are eliminated.

2). Repeating letters

Sometimes letters in a tweet are repeated to emphasize the emotion. For instance, hungrrrryyyy or huuuuungry could be used in the place of a simple 'hungry'. If a letter in a word repeats for two or more times, the repeating letters are replaced by exactly two of the same.

3). Ensuring words begin with an alphabet

With the concern of upholding simplicity, the words which do not begin with letters are eliminated. For example, 15th, 5.34 am are removed.

The feature vector extracted for the positive, neutral, and negative tweets are as below.

The feature words for the positive tweets shown above are as follows (in order):

- 'hey', 'cici', 'luv', 'new', 'iphone'
- 'heard', 'congrats','new','nexus'
- 'new', 'nexus', 'awesome', 'check'

The feature words for the neutral tweets shown above are as follows (in order):

- 'apple','iphone','plus','australian','pricing',availability','iphone','iphone6s'
- 'everything','nexus','appleevent','smartphones'
- 'thing','changed','iphone6s','apple'

The feature words for the negative tweets shown above are as follows (in order):

- 'new','iphone','heaviest','made','iphone6s','iphone6plus'
- 'smashed','iphone','sign', 'iphone6s' ,'shotoniphone6'
- 'nexus', 'expensive', 'new'

After the tweets have been subjected to the previous phases, a feature vector of the tweet results. Feature Vector is the most important concept in building a sentiment analyser. The success of the classifier is directly dependent on the construction of a good feature vector [3].

D. Model Building

1). Training

The polarity labelled data from corpus is parsed. Then the relevant features are extracted from it to construct the feature vector. This vector is in turn used to create a Feature List, which is a list of all the features of all the data items in dataset used for training. This list is stored in a text file on secondary memory for further use in both the training and classification [3].

2). Naive Bayes Classifier

Naive Bayesian Text Classification algorithm is used for classification after it has been trained. It uses a probabilistic approach to text classification. Here, the class labels are known and the goal is to create probabilistic models, which can be employed to classify new texts. It is specifically formulated for text and makes use of text specific characteristics. The Naive Bayesian classifier treats each document as a "bag of words" and the generative model makes the following assumptions: firstly, words of a document are generated independently of context, and, secondly, the probability of the word is independent of its position. This is why the name 'Naïve' was used for this algorithm. In real text documents the words often correlate with each other and the position of the word in text may play role. Multinomial Naive Bayes model is shown in the equation 1 [3].

$$P(c|d) := \frac{(P(c)\sum_{i=1}^{m} P(f|c)^{ni(d)})}{P(d)}. \qquad (1)$$

In this formula, f represents a feature and $n_i(d)$ represents the count of feature f_i 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 [3].

3). Classification

To classify a tweet, the following tasks are done. First, preprocessing of the tweet. Second, a feature vector of test data is formed. Third, the test data is inputted into the Naive Bayes classifier which has been trained by the training data. Finally, the Naive Bayes conditional probability formula is used to get polarity of the test tweet [3].

IV. Testing and Comparison

For training of the classifier, publicly available twitter datasets have to be used. These datasets consist of the tweet along with its sentiment specified. However, for testing the classifier, it is very important to use a new set of tweets that have not been previously encountered in training to test the efficiency of the classifier.

Tweets relating to a specific keyword are extracted from the Twitter API. These tweets fetched are no more than seven days old. The extracted tweets are now subjected to preprocessing and filtering to form the feature vector. The feature vector if fed into the trained classifier and consequently each inputted tweet is categorized as positive, negative, or neutral. This final output of the classifier is fed into the SAP Lumira software to obtain a graphical view of the cumulative results, that is, the total number of tweets that are positive, negative, and neutral for a specific product are depicted [2].

The unigram feature vector approach for tweet classification has been employed. This is the simplest and a very straightforward way to retrieve features from a tweet. The machine learning algorithms performance is quite average with this feature vector. To improve the performance of this classifier, a large training dataset needs to be used. Thus, the accuracy of performance of the Naïve Bayes Classifier is dependent on the size of the training dataset employed [2].

An alternative to the unigram approach is the bigram approach. To understand the bigram approach, consider an example here. For instance, 'not bad' (bigram) completely changes the sentiment compared to adding 'not' and 'bad' individually. Here, for simplicity, we have considered only unigrams [3].

For best accuracy, the unigram+bigrams must be used.

To conclude, the accuracy of classification can be increased by two means. First, by increasing the size of the training data set. Second, by using the unigrams and bigrams approach [3].

V. Scope and Future Work

We have used the unigrams approach to implement a sentiment analyzer using Naïve Bayes Classifier. A few interesting works that can be done are:

A. Working on different gram approaches

Research has shown that using of the unigram + bigram approach greatly improves accuracy of sentiment analysis of tweets. Hence, we intend to implement this approach.

B. Adding other datasets

We have only worked with the twitter dataset. This concept could be extended to other social media datasets like Facebook, LinkdedIn, and Google+, etc. [3].

C. Semantics

The classifier determined the holistic sentiment of a tweet. The polarity of a tweet depends on the perspective the tweet is interpreted from. For example, in the tweet “X beats Y :)”, the sentiment is positive for X and negative for Y. In this case, semantics plays a significant part. Using a semantic role labeller may indicate which noun is mainly associated with the verb and the classification would take place accordingly. This may allow “X beats Y :)” to be classified differently from “Y beats X :)” [1].

D. Bigger Dataset

We have using a training dataset of 20,000 tweets. However, using an extremely large training dataset will cover a better range of twitter words and hence a better unigram feature vector can be formed. This results in an overall improved model. Consequently, the accuracy of the classifier would be escalated [1].

E. Context Classification

Our model could be extended to detect sarcasm in tweets and meanings based on the context [3].

F. Language Options

Classification of tweets in foreign languages like Spanish, Hindi, Russian, etc., would be an interesting extension to this implementation [3].

G. Using Emoticon Data in Tweet Set

Emoticons are stripped from our training data. This implies that if our test data contains an emoticon feature, the classifier is not influenced towards a class. This should be addressed because the emoticons are vastly used in tweets and hence the emoticon features are very valuable [1].

VI. Results

The unigram feature vector is the simplest way to retrieve features from a tweet. The machine learning algorithms perform average with this feature vector. One of the reasons for the average performance might be the smaller training dataset of tweets. If one could get hold of millions of tweets and train these classifiers, the accuracy would improve substantially.

VII. Conclusion

Using a novel feature vector of unigrams, we have shown that machine learning algorithms such as Naïve Bayes achieve competitive accuracy in classifying tweet sentiment.

Acknowledgement

We would like to express our sincere gratitude to Dr.K.G.Srinivasan and all the faculty of the Computer Science and Engineering Department of M.S.Ramaiah Institute of Technology for encouraging us to pursue this project and also for guiding us throughout its undertaking.

References

[1] Ravikiran Janardhan, “Twitter Sentiment Analysis and Opinion Mining”.

[2] L.Hemalatha, Dr. G.P.Saradhi Varma, Dr. A. Govardhan, “Processing the Informal Text for Effective Sentiment Analysis”, International Journal of Emerging Trends and Technology in Computer Science, Volume 1, Issue 10, ISSN: 2278-6856.

[3] Dhiraj Ghurke, Niraj Pal, Rishit Bhatia, “Effective Sentiment Analysis of Social Media Datasets using Naïve Bayesian Classification”, International Journal of Computer Applications, Volume 99, No. 10.