

# Probabilistic User-level Opinion Detection on Online Social Networks

Kasturi Bhattacharjee and Linda Petzold

Department of Computer Science,  
University of California Santa Barbara.

**Abstract.** The mass popularity of online social networks such as Facebook and Twitter makes them an interesting and important platform for exchange of ideas and opinions. Accurately capturing the opinions of users from their self-generated data is crucial for understanding these opinion flow processes. We propose a supervised model that uses a combination of hashtags and n-grams as features to identify the opinions of Twitter users on a topic, from their publicly available tweets. We use it to detect opinions on two current topics: U.S. Politics and Obamacare. Our approach requires no manual labeling of features, and is able to identify user opinion with a very high accuracy over a randomly chosen set of users tweeting on each topic.

## 1 Introduction

Social networks have emerged as one of the most powerful means of communication today. From beginning as a medium through which people remained connected to friends and family, they have emerged to become a facilitator of social causes and revolutions. Facebook and Twitter proved to be an effective medium of communication for protesters during the Arab Spring, enabling them to coordinate and conduct a revolution [24, 21]. More recently, social media has been instrumental in facilitating the protests in Ukraine [27]. The massive popularity of social networks has led to their extensive use in political campaigns as well [34]. Social and political organizations such as *MoveOn.org* [15] and *Avaaz.org* [3] have emerged as platforms through which people start online petitions to increase public awareness on a myriad of social and political issues of importance.

Knowing the opinions of people is useful not only for predicting the outcome of socio-political events, but also for viral marketing, advertising and market prediction [20, 7, 2]. Since the volume of social network posts generated on a daily basis is enormous, it is important to be able to perform opinion detection in an automated fashion.

In this work we focus on the detection of opinions of Twitter users on a given topic by extracting informative features from their publicly available tweets, using a supervised learning approach. We chose two topics for which users tend to have strong opinions: U.S. Politics (during the 2012 Presidential Election) and Obamacare.

Twitter has gained popularity among researchers due to its emergence as one of the most widely used social networks, and also because it allows for the crawling of some of its data. However, this data also brings along with it a host of challenges. The short length of a tweet, the abundance of grammatical errors, misspelt words, informal language and abbreviations make it difficult to extract the opinion expressed through a tweet accurately.

To overcome the above issues, we adopt the following strategies. We begin by preprocessing the data to reduce the amount of noise as described in detail in Section 3. This is a non-trivial step especially when dealing with Twitter data. Because the opinions detected on the basis of a single *tweet* are unreliable, we focus instead on assessing the opinion of a *user* by aggregating the information in all of their tweets relating to the topic of interest over a given time period. We use a probabilistic approach, regularized to avoid overfitting [26], to classify the user opinions as positive or negative on a given topic. The selection of features is critical to this task. We found that combining the use of hashtags and n-grams was highly informative in detecting user opinion. It is to be noted here that our method requires no prior manual selection or labeling of features. To test the robustness of our methodology, we implemented it for the detection of political opinions on the 2012 U.S. Presidential election, and on the topic of Obamacare, and obtained a high level of accuracy.

The remainder of this paper is organized as follows. In Section 2 we discuss related research. Section 3 describes the Twitter data that we crawled, and the labeling of users for the training and test sets. In Section 4, we present the model and the features that we used for opinion detection. Section 5 describes the experiments conducted on our test dataset, and the results obtained. Finally, in Section 6, we present conclusions on our work.

## 2 Related Work

Research involving sentiment analysis or opinion mining on social networks may be divided into two areas: techniques that are based on lexicons of words, and techniques that are based on machine learning. The lexicon-based methods work by using a predefined collection (lexicon) of words, where each word is annotated with a sentiment. Various publicly available lexicons are used for this purpose, each differing according to the context in which they were constructed. Examples include the Linguistic Inquiry and Word Count (LIWC) lexicon [32, 31] and the Multiple Perspective Question Answering (MPQA) lexicon [25, 40, 41]. The LIWC lexicon contains words that have been assigned into categories and matches the input text with the words in each category [23]. The MPQA lexicon is a publicly-available corpus of news articles that have been manually annotated for opinions, emotions, etc. These lexicons have been widely used for sentiment analysis across various domains, not just specifically for social networks [1, 17, 8]. Other popular sentiment lexicons that have been designed for short texts are SentiStrength [38] and SentiWordNet [13, 4]. These lexicons have been extensively used for sentiment analysis of social network data, online posts, movie

reviews, etc. [16, 22, 37, 12]. However, as we will see in Section 5.3, they do not perform very well when applied to our problem of assessing user opinion.

Machine learning techniques for sentiment analysis include classification techniques such as Maximum Entropy, Naive Bayes, SVM [18], k-NN based strategies [11], and label propagation [39]. These usually require labeling of data for training, which is accomplished either by manually labeling posts [39], or through the use of features specific to social networks such as emoticons and hashtags [18, 11]. Some of the existing research combines lexicon-based methods and machine-learning methods [36]. These papers address a different (but related) problem than ours in that they perform tweet-level as opposed to user-level sentiment analysis. In Section 5.5, we will compare our method to user-level sentiment generated via tweet-level sentiment obtained by the methods of [36] and [18].

The methods in [35, 30, 9] perform user-level sentiment analysis. The method in [30] uses features derived from four different types of information of a social network user: user profile, tweeting behavior, linguistic content of the messages and the user network. Our method focuses on extracting informative features from only a user’s tweets, and can achieve high accuracies with a smaller number of features and a simpler model. The methods in [9] determine the political alignment of Twitter users using their tweets, as well as their retweet networks. The dataset is selected by first creating a set of politically discriminative hashtags that co-occur with the hashtags *#p2* (“Progressives on Twitter 2.0”) and *#tcot* (“Top Conservatives on Twitter”). The tweets selected for the dataset carry at least one of the discriminative hashtags. In contrast, we select our dataset via identification of users who use the generic keywords in Table 1 at least once, which does not require the determination of discriminative words or hashtags. Moreover, [9] does not conduct any study on using combinations of hashtags and n-grams as features, which we have found to yield the best performance in opinion detection across two different topics (as described in Section 5.4 of this paper). Thus the results are not directly comparable. In addition, our method performs automatic feature selection, which [9] does not address. In [35], user-level sentiment analysis is performed using the users’ following/mention network information. Since our dataset consists of randomly chosen users, we do not have the entire neighborhood of any user.

### 3 Datasets

**Data Collection.** We focused on two current topics for which people were more likely to voice their opinions on social media: U.S. Politics and Obamacare. For each of the topics of interest, we randomly selected users and collected their tweets over a period of time using the Twitter REST API. For U.S. Politics, our tweets were collected over the period of January 2012 to January 2013. The time period of the data collection coincided with the political campaigns leading up to the November 2012 U.S. Presidential election. For the dataset on Obamacare, we crawled tweets for 6 weeks over the months of June and July 2013.

To extract topical tweets, we filtered out tweets that contained words related to the topic of interest. For instance, for political tweets, we used words related to political figures, parties, causes or issues, or commentators whose bias is well-known. This approach is similar to that used in [33]. Table 1 shows the list of keywords used to obtain both the datasets and the categories that they belong to. The political dataset thus obtained was composed of 672,920 tweets from 552,524 users. The Obamacare dataset consisted of 187,141 tweets from 65,218 users. For the purposes of training and testing, we randomly picked users from each of the datasets, and then assigned a positive or negative opinion label (definition of these opinions are provided in Section 4.1) to them by manually reading *all* of their tweets. We labeled only those users whose opinion could be unambiguously determined from their tweets. We randomly chose 490 users (222 positive and 268 negative) for our labeled dataset on U.S. Politics, and 201 users (90 positive and 111 negative) for our labeled dataset on Obamacare.

Table 1: Keywords used to filter out topical tweets

Keyword	Keyword Type	Dataset
obama	Political figure	U.S. Politics
democrat	Political Party	U.S. Politics
p2	Political Party	U.S. Politics
romney	Political figure	U.S. Politics
gop	Political party	U.S. Politics
tcot	Political party	U.S. Politics
obamacare	Term for affordable health care	Obamacare
koch	Industrialists who are against Obamacare	Obamacare
affordable care	Term for affordable health care	Obamacare

**Data Cleaning and Preprocessing.** Twitter data is inherently noisy and filled with abbreviations and informal words. We performed the following cleaning and pre-processing on the dataset to enable a better extraction of features from it.

1. URL removal: In our method, URLs would not contribute to the feature extraction and were therefore removed.
2. Stop word removal: Stop words such as “a”, “the”, “who”, “that”, “of”, “has”, etc. were removed from the tweets before extracting n-grams, which is a common practice.
3. Punctuation marks and special character removal: Punctuation marks such as “:”, “,”, etc. and special characters such as “[ ]”, “””, “—”, etc. were removed before extracting n-grams.
4. Additional whitespace removal: Multiple white spaces were replaced with a single whitespace.
5. Conversion to lowercase: Tweets are not generally case-sensitive owing to the informal language used. For instance, for our method, the word “Obama” should be considered the same as “obama” when parsing through a tweet.

Hence we converted the tweets to lowercase to preserve uniformity in feature extraction.

6. Tokenization: The tweets were tokenized into words to extract n-grams from them. We use Python’s Natural Language Toolkit 3.0 [6] for this purpose.

## 4 Methods

Given a user’s tweets over time on a predetermined topic, our goal was to predict her opinion as accurately as possible. Thus we sought to learn a predictive model for user opinion from features derived from the tweets. In this section, we describe the problem definition, the model we used to solve the problem, and the features used for extracting user opinion. The results obtained are reported in Section 5.4.

### 4.1 Problem Definition

We adopted a probabilistic view for the user opinion in that we assumed it to be a distribution over *positive* and *negative* types. On the topic of US politics, we arbitrarily defined *positive* to mean that the user is pro-Obama or anti-Romney, and *negative* to mean that she is anti-Obama or Pro-Romney. On the topic of Obamacare, *positive* was again arbitrarily defined to be a pro-Obamacare opinion, and *negative* was defined to be an anti-Obamacare opinion.

The main challenges involved were: (1) to determine appropriate features that carry information about the user’s opinion (2) to learn a model that, with a sufficiently high accuracy, predicts the probabilistic user opinion from the features.

Thus, the problem definition may be summarized as follows: *Given a user’s tweets over time on a topic, we seek the probabilities of her having a positive or a negative opinion.*

### 4.2 Model

We cast the problem at hand as a supervised binary classification problem in which the classifier outputs the probabilities of the opinions that a user can have. Logistic regression is a well-known and widely used probabilistic machine learning tool for classification. Given a binary output variable  $y$  and a set of features  $X$ , logistic regression estimates the conditional distribution  $P(y = 1|X; \theta)$ , where  $\theta$  represents the parameters that determine the effect of the features on the output.

Logistic regression utilizes the following transfer function between  $X$  and  $y$ :

$$P(y = 1|X, \theta) = h_{\theta}(X) = \frac{1}{1 + \exp(-\theta^T X)}. \quad (1)$$

To estimate the parameter  $\theta$  of the logistic model, we use Maximum Likelihood Estimation. Assuming that we have  $m$  i.i.d training samples  $(y^i, X^i)$ ,  $i = 1, \dots, m$ , the log likelihood is given by

$$\log P(y|X, \theta) = \sum_{i=1}^m (y^i \log(h_\theta(X^i)) + (1 - y^i) \log(1 - h_\theta(X^i))). \quad (2)$$

The loss function, which is the negative log-likelihood, being convex, we can minimize it to estimate the optimum  $\theta$ , given by  $\hat{\theta}$ . We add a regularization to the loss function to avoid overfitting, as discussed in Sections 5.1 and 5.2.

Thus, given a set of features  $X$  and a set of known outputs  $y$  in the training data, the logistic regression model learns the parameter  $\theta$  that determines the relationship between  $X$  and  $y$ . Once the model has been learned, it can then be used to predict the outcomes of the test data, given their features  $X$ .

### 4.3 Features for Classification

Deriving features from the tweets is a crucial step for successfully determining a user’s opinion. The features must be such that they would reflect the opinion conveyed through the user’s tweets, because if a human annotator were to determine the opinion of a user (which is the baseline we are comparing with), she would read the user’s tweets to reach a conclusion.

Hashtags have become a very popular feature in Twitter and other social media sites. A hashtag is essentially a word that is prefixed with a # symbol that can be generated by a user and used in their tweets. *#followfriday*, *#mtvstars*, *#ipad*, *#glee* are examples of some popular hashtags on Twitter. The concept of hashtags was introduced in order to index tweets of a similar topic together, to make it easier for users to start a conversation with each other.

Apart from highlighting the topic of a tweet, hashtags have been found to carry some additional information regarding the bias of the tweet itself [11, 39]. For example, hashtags such as *#ISupportStaceyDash*, *#iloveapple*, *#twilightsucks* all carry information about the topic of the tweet and also clearly exhibit the bias of the user. A manual inspection of our dataset suggested that hashtags might be used to provide information about the bias of the tweet. For example, hashtags such as *#romneyshambles*, *#gopfail*, *#defundobamacare* were more likely to occur in tweets in which the user portrays a negative opinion towards the topic. Similarly, hashtags such as *#iloveobama*, *#istandwithobama*, *#getcovered* occurred most often with tweets that carried a positive opinion towards the respective topic. For this reason, our first choice for features to use was hashtags.

Although hashtags are powerful carriers of sentiment information, sometimes they may not be sufficient to convey the bias hidden in the tweet. For instance, hashtags may just refer to a political party without seemingly carrying any bias, in which case the information we seek may be carried by the text of the tweet. Here is an example of such a tweet:

*“@MittRomney’s refusal to release details of, well, anything, prove his cowardice & unfitness for the presidency. #connecttheleft #gop”*

In the above tweet, the hashtags used are *#gop* (“Grand Old Party”) and *#connecttheleft* (a hashtag designed to connect the Democrats). Used together, these hashtags carry no information on the user’s opinion. However, a human

annotator would be able to identify the opinion by reading the entire text of the tweet. Hence, in order to augment the information obtained by using hashtags alone, we incorporated information from the tweet as well.

For this purpose, we use the n-gram model which is considered a powerful tool for sentiment extraction [5]. n-grams are essentially contiguous sequences of  $n$  words extracted from text. The n-gram model was developed as a probabilistic language model which predicts the occurrence of the next word in the sequence of words by modeling it as an  $(n - 1)$ -order Markov process. In the domain of sentiment analysis, n-grams have been widely used since they help to capture phrases that carry sentiment expression [28, 10].

We begin by using hashtags separately as features in the logistic regression model (as described further), and then use them in conjunction with n-grams to achieve better results.

**Popular hashtags.** To eliminate the need for manual labeling of the hashtags, we extracted the most popular hashtags separately from each of the filtered datasets, by computing the total number of times each hashtag occurred in the respective dataset. For both the datasets, we used the 1000 most popularly used hashtags. We refer to these hashtags as *popular hashtags*. Not surprisingly, a manual inspection revealed that all of the popular political tags were related to politics either by representing names of the parties, their representatives, or political issues that gained importance during that time period. A similar pattern was observed for the popular Obamacare hashtags.

We then used the frequency of use of the popular hashtags as features in our model. Thus, in equation (3),

$$X_j^i = \text{number of times popular hashtag } j \text{ is used by user } i. \quad (3)$$

**Popular n-grams in conjunction with hashtags.** As discussed previously, we used n-grams to augment the hashtag information. We used values of  $n = 1, 2$  to extract out unigrams and bigrams from the tweets of each labeled user. Again, we picked the most popular n-grams from each dataset. For each dataset, we chose 2000 most popular unigrams and 2000 most popular bigrams. We combined the information we obtained from the hashtags with that obtained from the n-grams. This was done by performing logistic regression using multiple explanatory variables as follows

$$P(y = 1|X, Z, \theta, \beta) = \frac{1}{1 + \exp(-\theta^T X - \beta^T Z)}, \quad (4)$$

where  $X$  and  $Z$  represent the hashtag-based features and the n-gram-based features respectively;  $\theta$  and  $\beta$  represent the corresponding parameters. We tested each *type* of n-gram feature separately with the hashtags.

## 5 Experimental Results

In this section we outline in detail implementations of the proposed method with both  $l_1$  and  $l_2$  regularization, and the metrics we used to evaluate the results.

Further, we describe the existing methods that we chose for comparison, and report the results obtained.

### 5.1 Logistic regression with $l_2$ regularization

Using both hashtags and n-grams yields a relatively large number of features (3000 for hashtags and bigrams). To avoid overfitting, we add a user-specified regularization term  $\lambda\|X\|_2^2$  to our loss function, where  $\lambda > 0$  is the regularization parameter [26]. The loss function thus becomes:

$$L(\theta) = -\log P(y|X, \theta) + \lambda\|\theta\|_2^2. \quad (5)$$

### 5.2 Logistic regression with $l_1$ regularization

We also explored the use of  $l_1$ -regularization [26]. This results in the loss function:

$$L(\theta) = -\log P(y|X, \theta) + \lambda\|\theta\|_1. \quad (6)$$

We used the open-source machine learning tool in Python, scikit-learn [29] to implement logistic regression with  $l_1$  and  $l_2$  regularizations. The selection of  $\lambda$  is discussed in Section 5.3.

Table 2: Metrics using  $l_2$ -regularization on U.S. Politics dataset

Feature type	Total Number of Features	Number of Selected Features	Mean Accuracy	Mean AUC	Mean F1-score	Mean Specificity
Popular hashtags	1000	288	86.32( $\pm$ 0.043)	0.915	0.85	0.875
Popular hashtags, unigrams	3000	1488	86.12( $\pm$ 0.031)	0.896	0.843	0.885
Popular hashtags, bigrams	3000	1398	<b>87.35(<math>\pm</math>0.029)</b>	<b>0.909</b>	<b>0.858</b>	<b>0.895</b>
Popular hashtags, unigrams, bigrams	5000	2430	87.10	0.905	0.855	0.893

### 5.3 Evaluation metrics

To evaluate the performance of the model, we conducted hold-out cross validation by randomly splitting the data into 30% test set and 70% training set. On



each run of the cross-validation, the best  $\lambda$  was learned from the validation error on the training set. The cross-validation was done 10 times, with the data being randomly shuffled each time. Our experiments showed that the best  $\lambda$  value did not vary much across the validation sets of the respective dataset. For the U.S. Politics dataset, we set  $\lambda = 50.0$  for  $l_2$ -regularization, and for  $l_1$ -regularization, it was 0.01. For the Obamacare dataset, we set  $\lambda = 25.0$  for the  $l_2$ -regularized model, and  $\lambda = 0.0083$  for the  $l_1$ -regularized model. The average classifier metrics [14] such as ROC curves, AUC, accuracy, precision, recall, F1-score and specificity across the 10 sets is reported in Section 5.4. For the U.S. politics data, we tested on 147 users, and on 60 users for the Obamacare dataset. For each user, the class with the higher probability is assigned as the corresponding opinion label, with ties broken arbitrarily. There were no cases in either of the datasets in which ties were encountered.

#### 5.4 Results

Table 3: Metrics using  $l_2$ -regularization on Obamacare dataset

Feature type	Total Number of Features	Number of Selected Features	Mean Accuracy	Mean AUC	Mean F1-score	Mean Specificity
Popular hashtags	1000	445	77.33( $\pm 0.0466$ )	0.912	0.804	0.799
Popular hashtags, unigrams	3000	2295	87.30( $\pm 0.022$ )	0.942	0.906	0.943
Popular hashtags, bigrams	3000	1506	87.54( $\pm 0.025$ )	0.956	0.907	0.927
Popular hashtags, unigrams, bigrams	5000	3448	<b>90.8 (<math>\pm 0.033</math>)</b>	<b>0.958</b>	<b>0.919</b>	<b>0.850</b>

Table 2 and Figure 1(a) present the results obtained using logistic regression with  $l_2$  regularization on U.S. Politics, and Table 3 and Figure 1(b) demonstrate the results on the Obamacare dataset. We ran this method using four combinations of features, as shown in the results. As can be observed, the values of each of the classifier metrics are excellent. The high values of precision and specificity indicate that the method could predict both positive and negative opinions accurately. The highest accuracy achieved by our classifier was **87.35%** on U.S. Politics and **90.8%** on Obamacare. Figure 1(c) presents the ROC curves obtained using the  $l_2$ -regularized model on U.S. Politics and Obamacare.

Table 4 presents the results obtained with  $l_1$ -regularized logistic regression using the same kinds of features on the U.S. Politics dataset. It is to be noted that, using  $l_1$  regularization, comparable accuracies were obtained with a much smaller number of features. For instance, using the combination of hashtags and bigrams, we were able to achieve a high accuracy of **86.10%** and an AUC of **0.916** from 32 features, as contrasted with using 1398 features and obtaining slightly higher accuracy of 87.35% and an AUC of 0.909 with  $l_2$  regularization. Similarly, Table 5 shows the results using  $l_1$ -regularized logistic regression on the Obamacare dataset. A similar trend in results is observed in this case as well.

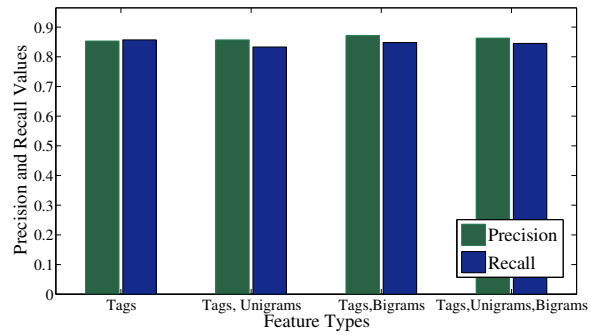
**Selection of Informative Features.** From Tables 4 and 5, we find that the  $l_1$  regularizer yields excellent results with a small number of selected features. Table 6 shows a few of the features that the regularizer picked from either dataset as the most informative features. Thus the method results in automatic selection of the most useful features for opinion detection.

### 5.5 Comparison with existing methods

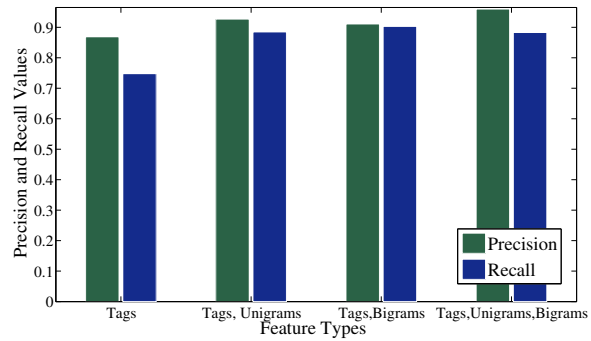
We compare our methods with three popularly used state-of-the-art methods that perform tweet-level sentiment analysis, and use their results to obtain opinion on a user level as described below. The following methods were tested on the U.S. Politics dataset.

**SentiStrength.** SentiStrength [38] is a lexicon-based method that was designed for use with short informal text including abbreviations and slang. It has been widely used by researchers for sentiment analysis of tweets, online posts, etc. (Section 2). It uses a lexicon of positive and negative words which were initially annotated by hand, and later improved during a training phase. Given a sentence, the method assigns a sentiment score to every word in the sentence, and thereafter, the sentence is assigned the most positive score and the most negative score from among its words. According to [38], the algorithm was tested extensively for accuracy, and was found to outperform standard machine learning approaches. Hence we chose this as a baseline method to compare against.

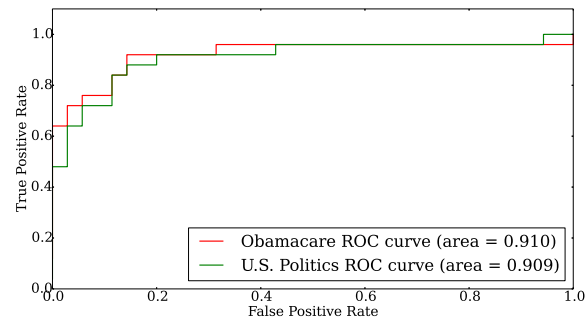
**Tweet-level Maximum Entropy classifier.** The second method for comparison is a machine-learning method proposed in Section 3.3 of [18] which uses a Maximum Entropy based classifier trained on 1,600,000 tweets using emoticons as noisy labels. It uses the presence or absence of unigrams, bigrams and parts-of-speech tags as features for classification, and classifies a given tweet as positive or negative. The authors provide an online tool for this purpose [19], which we use for conducting our experiments. This method has also been widely used for sentiment analysis. It is to be noted that we used their pre-trained model that was trained on their annotated tweet set. We could not train the method on our labeled datasets because our datasets have labels on the user and not on the individual tweets, and it is non-trivial to transfer the user opinion to their tweets owing to the amount of noise per tweet. Moreover, we could not annotate *our* datasets using emoticons because they are rarely used in our datasets (only 0.13% of the tweets used emoticons in the U.S. Politics dataset). Since they used



(a) Precision Recall values for U.S. Politics



(b) Precision Recall values for Obamacare



(c) ROC curves for U.S. Politics and Obamacare

Fig. 1: Classifier metrics with  $l_2$  regularization

Table 4: Metrics using  $l_1$  regularization on U.S. Politics dataset

Feature type	Total Number of Features	Number of Selected Features	Mean Accuracy	Mean AUC	Mean F1-score	Mean Specificity
Popular hashtags	1000	22	84.70( $\pm$ 0.048)	0.896	0.823	0.82
Popular hashtags, unigrams	3000	34	85.67( $\pm$ 0.025)	0.903	0.818	0.86
Popular hashtags, bigrams	3000	32	<b>86.10(<math>\pm</math>0.030)</b>	<b>0.916</b>	<b>0.844</b>	<b>0.849</b>
Popular hashtags, unigrams, bigrams	5000	70	85.03	0.909	0.832	0.869

emoticons to label the sentiment of a tweet and did not manually annotate them, theirs may be considered as a (partially) supervised method, as opposed to our fully supervised method.

**Combined Method.** The third method for comparison is a method described in Section 3.2 of [36] that combines the output of the lexicon-based method [38] and the tweet-level machine learning method [18]. The authors propose a way to combine the results of SentiStrength and the MaxEnt based method of [18] to perform a binary tweet-level sentiment classification with better accuracy than either of the individual methods.

**Obtaining targeted user-level sentiment from tweet-level sentiment.**

We adopt the following strategies when comparing our method with the other three methods. First, to obtain a sentiment label for every tweet using *SentiStrength*, the most positive and most negative scores for every tweet were added up. If this sum was positive the tweet was labeled positive; if the sum was negative then it was labeled negative, and if the sum was zero the tweet was labeled neutral. This approach was proposed in Section 3.2 of [36].

Second, all of the methods described above determine whether a given tweet has an *overall positive or negative sentiment, irrespective of the target of the sentiment*. This varies from our definition of *positive* and *negative* as described in Section 4.1. Hence, to determine the sentiment of a tweet towards a target (Democrat or Republican), we selected a set of keywords that were associated with Democrats and another set for Republicans, with the objective of identifying targets for as many tweets as possible, and defined them as *positive targets* and *negative targets*, respectively. (The keywords used are given in Table 7). For **any** method that we compared with, given a tweet sentiment, we first computed a sum of the target words that the tweet contained, assigning +1 for a *positive target* and -1 for a *negative target*. If the sum was greater than 0 we assumed

Table 5: Metrics using  $l_1$  regularization on Obamacare dataset

Feature type	Total Number of Features	Number of Selected Features	Mean Accuracy	Mean AUC	Mean F1-score	Mean Specificity
Popular hashtags	1000	34	77.33( $\pm 0.0466$ )	0.912	0.804	0.799
Popular hashtags and unigrams	3000	210	87.30( $\pm 0.022$ )	0.942	0.906	0.943
Popular hashtags and bigrams	3000	132	87.54( $\pm 0.025$ )	0.956	0.907	0.927
Popular hashtags, unigrams, bigrams	5000	372	<b>90.8 (<math>\pm 0.033</math>)</b>	<b>0.958</b>	<b>0.919</b>	<b>0.850</b>

that the subject of the tweet was Democrats, in which case the sentiment remained unaltered. If the sum was less than 0 we assumed that the subject was Republicans. In this case, a positive sentiment towards Republicans would mean a *negative* sentiment according to our definition, and vice versa.

Table 6: Examples of features selected by  $l_1$ -regularization

Feature Type	Dataset	Sparse features
Hashtags	U.S. Politics	“tcot”, “p2”, “gop”, “obama2012”
Bigrams	U.S. Politics	“tcot gop”, “obama didnt”, “mitt romney”
Hashtags	Obamacare	“obamacare”, “tcot”, “defundobamacare”, “defund”
Bigrams	Obamacare	“defund obamacare”, “shut down”, “government over”

Third, to obtain *user-level* sentiment from the tweet-level sentiment output from **any** of the methods, we adopted the following strategy. For every user, we summed the (targeted) sentiments of all her tweets using +1 for *positive*, -1 for *negative* and 0 for neutral. The user output was considered positive if the sum was positive, negative if the sum was negative and was assigned randomly if the sum was zero. Table 8 represents the comparison of our method with the existing methods. All of the classifier metrics clearly display that our method outperforms all the three methods.

Table 7: Keywords used to identify positive and negative targets in the U.S. Politics Dataset

Target Type	Keywords
Positive targets	“obama”, “democrat”, “p2”, “barackobama”, “barack”, “democrats”, “liberals”, “obama2012”, “dem”, “p2b”, “biden”, “romneyshambles”, “clinton”, “releasethereturns”, “forward2012”, “obamabiden2012”, “connecttheleft”, “ctl”, “inobamasamerica”, “obamawinning”, “dnc”, “dncin4words”, “dnc2012”, “150dollars”, “repugnican”
Negative targets	“romney”, “gop”, “tcot”, “mitt”, “mittromney”, “republicans”, “teaparty”, “imwithmitt”, “mitt2012”, “nobama2012”, “romneyryan2012”, “tlot”, “webuiltit”, “teaparty”, “gop2012”, “prolife”, “romneyryan”, “you didnt build that”, “obamaphone”, “anndromney”, “obamafail”, “you just pulled a romney”, “nobama”, “republican”, “limbaugh”, “paulryanvp”

Table 8: Comparison of the proposed method with three state-of-the-art methods

Method	Accuracy(%)	Precision	Recall	Specificity
$l_2$ - regularized Logistic regression	<b>87.35</b>	<b>0.871</b>	<b>0.848</b>	<b>0.895</b>
SentiStrength	53.06	0.485	0.586	0.485
Maximum Entropy method	44.29	0.525	0.419	0.463
Combined method (SentiStrength and MaxEnt)	59.59	0.542	0.694	0.515

## 6 Conclusion

In this paper we propose a method for detecting user-level opinion on a given topic from Twitter data. Our approach of performing user-level (as opposed to tweet-level) opinion detection using regularized logistic regression with hashtags and n-grams as features was found to produce excellent results. The  $l_2$  and  $l_1$  regularizations yielded comparable accuracy, however the  $l_1$  regularization required far fewer features. Moreover, our method required no manual labeling of features. The method was applied to Twitter datasets on two different topics and yielded excellent results on both, which highlights its generalizability. The importance of informative features is evident in the results obtained; only a small percentage of the most informative features were required for accurate user opinion detection.

## References

1. Akkaya, C., Wiebe, J., Mihalcea, R.: Subjectivity word sense disambiguation. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1. pp. 190–199. Association for Computational Linguistics (2009)

2. Asur, S., Huberman, B.A.: Predicting the future with social media. In: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. vol. 1, pp. 492–499. IEEE (2010)
3. Avaaz:: 2013. <http://www.avaaz.org/en/highlights.php> (The World in Action)
4. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: LREC. vol. 10, pp. 2200–2204 (2010)
5. Bespalov, D., Bai, B., Qi, Y., Shokoufandeh, A.: Sentiment classification based on supervised latent n-gram analysis. In: Proceedings of the 20th ACM international conference on Information and knowledge management. pp. 375–382. ACM (2011)
6. Bird, S., Klein, E., Loper, E.: Natural language processing with Python. ” O’Reilly Media, Inc.” (2009)
7. Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. *Journal of Computational Science* 2(1), 1–8 (2011)
8. Bono, J.E., Ilies, R.: Charisma, positive emotions and mood contagion. *The Leadership Quarterly* 17(4), 317–334 (2006)
9. Conover, M.D., Gonçalves, B., Ratkiewicz, J., Flammini, A., Menczer, F.: Predicting the political alignment of twitter users. In: Privacy, security, risk and trust (passat), 2011 IEEE Third International Conference on Social Computing (Social-Com). pp. 192–199. IEEE (2011)
10. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th international conference on World Wide Web. pp. 519–528. ACM (2003)
11. Davidov, D., Tsur, O., Rappoport, A.: Enhanced sentiment learning using twitter hashtags and smileys. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. pp. 241–249. Association for Computational Linguistics (2010)
12. Denecke, K.: Using sentiwordnet for multilingual sentiment analysis. In: Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on. pp. 507–512. IEEE (2008)
13. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC. vol. 6, pp. 417–422 (2006)
14. Fawcett, T.: An introduction to ROC analysis. *Pattern recognition letters* 27(8), 861–874 (2006)
15. Galland, A.: Moveon.org. <http://front.moveon.org/thank-you-for-an-awesome-2013/#.Uty0RnmttFQ> (2013)
16. Garas, A., Garcia, D., Skowron, M., Schweitzer, F.: Emotional persistence in online chatting communities. *Scientific Reports* 2 (2012)
17. Gilbert, E., Karahalios, K.: Predicting tie strength with social media. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 211–220. CHI ’09, ACM, New York, NY, USA (2009), <http://doi.acm.org/10.1145/1518701.1518736>
18. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford pp. 1–12 (2009(a))
19. Go, A., Huang, L., Bhayani, R.: Twittersentiment. <http://www.sentiment140.com> (2009(b))
20. Jansen, B.J., Zhang, M., Sobel, K., Chowdury, A.: Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology* 60(11), 2169–2188 (2009)

21. Kassim, S.: How the arab spring was helped by social media. <http://www.policymic.com/articles/10642/twitter-revolution-how-the-arab-spring-was-helped-by-social-media> (2012)
22. Kucuktunc, O., Cambazoglu, B.B., Weber, I., Ferhatosmanoglu, H.: A large-scale sentiment analysis for yahoo! answers. In: Proceedings of the fifth ACM international conference on Web search and data mining. pp. 633–642. ACM (2012)
23. LIWC: LIWC software. <http://www.liwc.net/index.php> (2001)
24. Marzouki, Y., Oullier, O.: Revolutionizing revolutions: Virtual collective consciousness and the arab spring. [http://www.huffingtonpost.com/yousri-marzouki/revolutionizing-revolution\\_b\\_1679181.html](http://www.huffingtonpost.com/yousri-marzouki/revolutionizing-revolution_b_1679181.html) (2012)
25. MPQA: MPQA. <http://mpqa.cs.pitt.edu/lexicons/> (2005)
26. Ng, A.Y.: Feature election,  $l_1$  vs.  $l_2$  regularization, and rotational invariance. In: Proceedings of the Twenty-First International Conference on Machine Learning. p. 78. ACM (2004)
27. Onuch, O.: Social networks and social media in ukrainian “euromaidan” protests. <http://www.washingtonpost.com/blogs/monkey-cage/wp/2014/01/02/social-networks-and-social-media-in-ukrainian-euromaidan-protests-2/> (2014)
28. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: LREC (2010)
29. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, 2825–2830 (2011)
30. Pennacchiotti, M., Popescu, A.M.: A machine learning approach to twitter user classification. In: ICWSM (2011)
31. Pennebaker, J.W., Chung, C.K., Ireland, M., Gonzales, A., Booth, R.J.: The development and psychometric properties of liwc2007. Austin, TX, LIWC. Net (2007)
32. Pennebaker, J.W., Francis, M.E., Booth, R.J.: Linguistic inquiry and word count: LIWC 2001 (2001)
33. Romero, D.M., Meeder, B., Kleinberg, J.: Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In: Proceedings of the 20th international conference on World wide web. pp. 695–704. ACM (2011)
34. Rutledge, P.: How obama won the social media battle in the 2012 presidential campaign. <http://mprcenter.org/blog/2013/01/25/how-obama-won-the-social-media-battle-in-the-2012-presidential-campaign/> (2013)
35. Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., Li, P.: User-level sentiment analysis incorporating social networks. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1397–1405. ACM (2011)
36. Tan, S., Li, Y., Sun, H., Guan, Z., Yan, X., Bu, J., Chen, C., He, X.: Interpreting the public sentiment variations on twitter. *IEEE Transactions on Knowledge and Data Engineering* (2012)
37. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment in twitter events. *Journal of the American Society for Information Science and Technology* 62(2), 406–418 (2011)
38. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology* 61(12), 2544–2558 (2010)



39. Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: Proceedings of the 20th ACM international conference on Information and knowledge management. pp. 1031–1040. ACM (2011)
40. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. *Language resources and evaluation* 39(2-3), 165–210 (2005)
41. Wilson, T.: Fine-grained subjectivity analysis. Ph.D. thesis, Doctoral Dissertation, University of Pittsburgh (2008)