




An Approach to Build a Sentiment Analyzer: A Survey

Singh Dharmendra , Bhatia Akshay , and Singh Ashvinder 

Department of Computer Science, Atma Ram Sanatan Dharma College,
University of Delhi, New Delhi, India
Singh88dk@gmail.com, Akshaybhatia95@gmail.com,
Ashvindersinghdhariwal@gmail.com

Abstract. With the increase in the use of social networking sites like Twitter anyone can share or express his or her views with each other on a common stage. Twitter sentiment analyzer is a tool which is used to find out whether a corpus of data is positive, negative or neutral. Our work focuses on the steps involved in this Opinion Mining problem necessary to fetch opinions out of a corpus. We also aim to look at the strengths and scope for future research in the field of Twitter sentiment analyzer.

Keywords: Opinion mining · Sentiment analysis · Twitter · Social networks

1 Introduction

Sentiment Analysis is a methodology that is involved in taking out the sentiments (positive, negative and neutral in this article). People buy things, read articles, watch movies, generating large amounts of data and thus expressing their opinions. Expressing people's opinion about a particular topic or product has become necessary for product based companies especially e-commerce websites to augment the user experience. Doing this manually is a very tedious task. To make the task easier, sentiment analyzers have been developed to parse opinions on social media. Twitter is a powerful medium where people share their experiences using hashtags which can, in turn, be used to find data about any particular topic and people's opinion is mined.

The paper is defined as follows – Sect. 2 of our report gives a brief explanation of related work in the field of opinion mining. Section 3 deals with general concepts involved in building a sentiment analyzer and specify the rule based approach and machine learning approach. It gives an overview of machine learning algorithms such as Naïve Bayes and Linear Support Vector Machines which are the most popular machine learning algorithms used in sentiment analysis. Section 4 describes the steps involved in making a sentiment analyzer that derives its corpus from Twitter, the famous social networking website. Section 5 deals with the applications/future scope of sentiment analysis. Section 6 concludes the paper.

2 Background Work

There has been much research in the field of opinion mining on Twitter and has shown promising application. In this domain, the biggest challenge is to translate and compute

the abbreviations and slangs used in the social networking world. Use of Twitter as a corpus for opinion mining has been cited by Pak and Paroubek [1]. They have used a three-way classification method which classifies the tweets into positive, negative and neutral. Kumar et al. [2] augmented this approach by using five emotions namely happiness, anger, fear, sadness and disgust. Hasan et al. [3] have used a Naïve Bayes based approach to design a classifier in both English and Bangla. Bhatia et al. [4] have presented a system flow model in opinion mining which points the crucial steps that a text should go through to get a trustworthy opinion from the internet. Balahadia et al. [5] have presented a Teacher Evaluation Architecture which can be used to measure not only the quantitative aspects but also the qualitative aspects of reviews from students to evaluate their Teachers. Fang et al. [9] have used opinion mining techniques on a set of data obtained from Amazon. They have used sentiment analysis to give product reviews.

3 Approaches Used in Building a Sentiment Analyzer

Two different approaches are involved in the construction of a sentiment analyzer:

3.1 Rule-Based Approach

Use of human judgment to find a relationship between sentences, sorting of words as positive or negative and manually doing lexical analysis is known as a rule-based approach. The major steps in rule-based approach consist of:

1. Identifying every word in the document as positive or negative
2. If the number of positive words is more than negative words, then classify the document as positive and vice versa.

VADER: Vader is a tool which uses rule-based methods to do opinion mining work on social media text.

A lexicon and rule-based sentiment analysis tool, i.e., VADER (Valence Aware Dictionary and sentiment Reasoner) is designed to manage the sentiments expressed on the social networks. It is sensitive both to the difference and the intensity of the sentiments voiced in the social networks. It is a cumbersome task to create sentiment lexicon manually and can be erroneous as well, hence opinion mining researchers are using already existing lexicons as their resource.

The below example shows how VADER sentiment is used in one's application.

- | | |
|--------------------------------------|---------------------------|
| 1. "Mohan is smart and intelligent." | Positive sentence example |
| 2. "Mohan is smart and intelligent!" | Punctuation emphasis |

VADER can analyze the inputs provided into negative, neutral and positive. It also provides a compound value which lies between -1 (Extremely negative) and $+1$ (Extremely positive).

Mohan is smart and intelligent.
 {'neg': 0.0, 'neu': 0.244, 'pos': 0.756, 'compound': 0.8326}
 Mohan is smart and intelligent!
 {'neg': 0.0, 'neu': 0.247, 'pos': 0.752, 'compound': 0.8439}
 Mohan is VERY SMART and INCREDIBLY INTELLIGENT!!!
 {'neg': 0.0, 'neu': 0.284, 'pos': 0.716, 'compound': 0.9469}

3.2 Machine Learning Based Approach

Machine Learning based approach is used to classify a document as positive or negative. There are many Machine Learning based algorithms, two of which are the Naïve Bayes Classifier and the Support Vector Machine. We will discuss these shortly but first, let us have a look at the steps to be followed in an ML-based approach –

Input – Input can include any documents or text. Here the ML-based classifier gets trained on the corpus of pre-classified texts. Once the classifier gets trained, it can easily classify the document as positive or negative as is shown in Fig. 1.

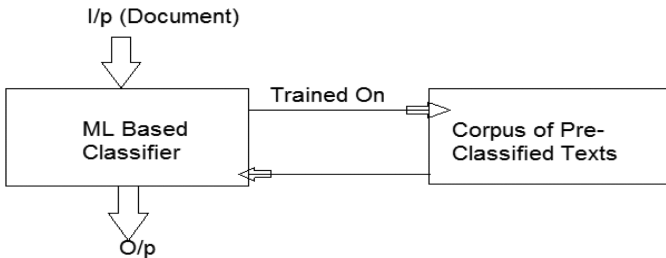


Fig. 1. Flowchart of a basic sentiment analyzer.

Several human annotated corpora are available that are used as training data. These corpora are already marked as positive or negative. They are used to train the ML-based classifier.

Training data include two columns – Text column and a label column which classifies a text as positive or negative (See Table 1)

Table 1. Columns of training data

Text	Label
Pride	Positive
Annoyed	Negative
.	Positive
.	.
.	.
.	.

Now, convert the trained text into numerical attributes.

e.g. “I Love Delhi” is converted to (3,0,1). Where the first attribute signifies a total number of words, the second one tells the number of negative words and the third one tells the number of positive words. This way features are developed which include all the numerical attributes.

Now, features and labels are added to the algorithm to train it so that it can further be used for analysis of new corpus as is shown in Fig. 2.

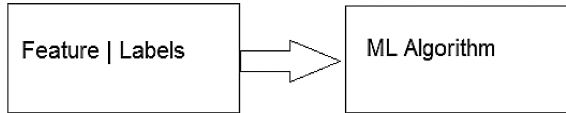


Fig. 2. Features and Labels are fed to a machine learning algorithm.

Finally the new text can now be fed to the algorithm which converts it into a positive or negative entity.

Overview of Naïve Bayes Classification: This is a machine learning based approach used to categorize a document as positive or negative. The simplest way to do so is to look at the individual words in the document.

With this algorithm, we compute the posterior probability. Given some evidence, we compute the probability that the document is positive or negative. Here the evidence is the words present in the document, so we will compute the conditional probability given the words in the document.

$$P(\text{Document is Positive}|\text{Words}) = \frac{P(\text{document is positive}) * P(\text{Word 1}|\text{document is positive}) * P(\text{word 2}|\text{document is positive}) * \dots}{P(\text{word 1}) * P(\text{word 2}) * \dots} \tag{1}$$

$$P(\text{Document is Negative}|\text{Words}) = \frac{P(\text{document is negative}) * P(\text{Word 1}|\text{document is negative}) * P(\text{word 2}|\text{document is negative}) * \dots}{P(\text{word 1}) * P(\text{word 2}) * \dots} \tag{2}$$

Here, P (document is positive) or P (document is negative) are prior probabilities that the document is positive or negative which is number of positive or negative documents in entire corpus divided by total number of documents. We multiply the prior probability with the conditional probability of words given that document is positive or negative and we multiply the probability of each word given that the words are independent of each other.

This is an assumption which is not true for most real life problems, but it has been proved that this algorithm works well with most classification problems. No attention is being given to the denominator which is the probability of a word occurring in a document. This is because we only need to know which of these probabilities is greater for the document. This classifier can also be used to classify the document into multiple classes like positive, negative or neutral.

Overview of Linear Support Vector Machine: Support Vector Machine or SHM is a supervised machine learning classifier which unlike Naïve Bayes is not probabilistic. SHM is binary and thus can only classify text as positive or negative, good or bad. It

cannot introduce a third category. For example, the text can be categorized as positive, negative or neutral. In LINEAR SHM, we cannot introduce this third category.

Also, the SHM makes a decision on the basis of a linear function of the point's coordinates. If a point is

$$X = (X_1, X_2, X_3 \dots \dots X_n) \tag{3}$$

Then a linear equation would look like

$$F(X) = aX_1 + bX_2 + cX_3 + \dots + zX_n \tag{4}$$

where a, b, c...z are constants that play a key role in classifying.

Then the support vector machine would run a test:

```
If F(x)>0 then the sentence is positive
Else negative;
```

The linear equation above represents an N-1 dimensional hyperplane in an N-dimensional Hypercube space. SHM does not involve any assumptions about probability distributions of the points involved, unlike Naïve Bayes. SHM involves a training stage when the model “learns” from a set of training data (Fig. 3).

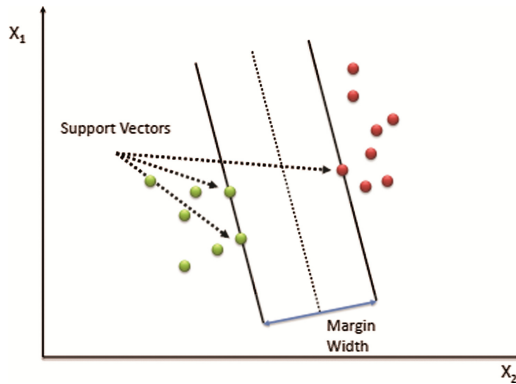


Fig. 3. Support vectors being separated by a hyperplane [7].

One has a bunch of corpus text that is already marked as negative and positive. Take these words and represent them as points in N-dimensional space. The hyperplane neatly separates the two cluster of points marked as positive or negative.

The hyperplane is the linear function that the SHM tries to find. This hyperplane can then be used to classify new data as positive or negative.

How SHM Constructs this Hyper-Plane?

The equation of the set of points on a hyperplane is always Linear. For three-dimensional space, the equation is

$$A.x + B.y + C.z = D \quad (5)$$

All points on the plane will satisfy the equation.
All points on one side would satisfy the condition,

$$A.x + B.y + C.z > D \quad (6)$$

Moreover, the other side would satisfy,

$$A.x + B.y + C.z < D \quad (7)$$

There might be many planes that divide the points into a set of two clusters, but there is only one specific plane that the SHM identifies. It depends on the distance of the point from the hyperplane.

Let a set of points be (x_1, y_1, z_1) then distance from hyperplane is calculated as:

$$\frac{Ax_1 + By_1 + Cz_1 - D}{(A^2 + B^2 + C^2)^{1/2}} \quad (8)$$

The best hyperplane is found by maximizing the sum of the distances of the nearest points on either side while making sure that two clusters are on opposite sides. The plane found is called the maximum margin hyperplane. Support vectors are simply the nearest points on either side of the hyperplane.

4 Twitter Sentiment Analysis Approach

The purpose of a Twitter sentiment analyzer is to accept data from Twitter as input and find the current sentiment on that input. We have used the machine learning based approach.

Accepting a Search Term and Downloading a Certain Number of Tweets for that Term:

Step1: (I) Twitter provides output in a difficult interpreting manner, therefore it is important first to convert it into a readable form using a language like Python.

(II) First, create a new application using the Twitter API that is registered with Twitter. API credentials are provided that are utilized in the program to provide authenticity.

Step2: Create a function whose argument is the search string that the user inputs. The string is used as a keyword to fetch tweets. Python has an inbuilt function API. Getsearch (String search, count = 100) that takes in the string and the number of tweets you want to your data.

Step3: Classify the 100 tweets as positive or negative manually or use an already available corpus to use as TRAINING DATA. Save the tweets into a Comma Separated Variable file. If one is using a pre-defined corpus, then one will need to process it further as Twitter will only provide tweet IDs and not the tweet text. To extract Tweet text, one will need to write a function that will read the Comma Separated Variable

and loop through each tweet ID downloading the corresponding tweet text from Twitter and gain writing it to another CSV file.

Pre-Processing the Tweet Text

Step1: Create a class to preprocess all the tweets, which can be used for both training and test data. We used regular expressions (Python regex) and NLTK (python natural language toolkit) for preprocessing.

Step2: Write a function that removes stopwords, for instance, a, an, the which do not contribute to the polarity of the text. These stop-words are recognized using NLTK in Python. Also, it will tokenize the tweets word by word.

Step3: Define a function that processes the tweets as follows:
 Convert all terms to lower case. Example: BEAUTY to beauty.

Substitute the links if any in the tweet with the word URL.

Replace @username with “AT_USER. “Remove both URL and “AT_USER” as they are stop-words using the function defined above.

Remove hashtags with only the text following hashtags. For example, #sunny is replaced with sunny. Figure 4 Shows the possible combination of # removal regex.

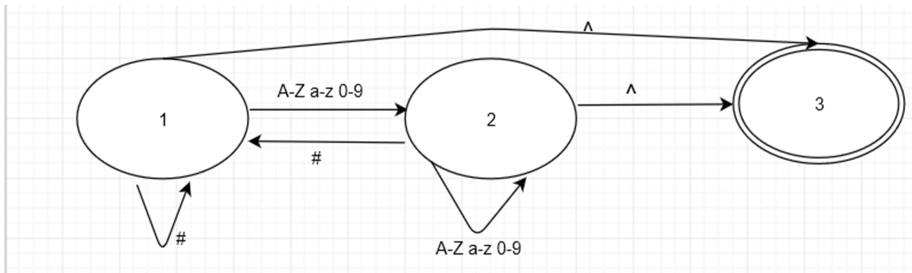


Fig. 4. Possible combination of # removal regex

Using A Machine Learning Classifier:

Step1: Extract features for both the test and training data. Do experimentation using different data.

Step2: Train a classifier on training data.

Step3: Use the classifier to classify the problem instances as positive or negative.

Naïve Bayes Feature Vector:

Step1: Build a vocabulary which is the list of all words in training data.

Step2: Represent each tweet with presence/absence of these words in the tweet.

For example: “Your Pasta is best in the world.” is a sentence in vocabulary.

“Pasta is best.” is a tweet. The feature vector for Naïve Bayes would be

[0, 1, 1, 1, 0, 0, 0], Here 1 is used to represent data that is there in the vocabulary and 0 is used to denote absence of that word.

Step3: Use NLTK built-in classifier to train data.

SVM Feature Vector:

The first two steps are the same as Naïve Bayes.

Step3: Weight each word with SentiWordNet [6] subjectivity score which is a lexical resource for opinion mining used for various research and industrial purposes. It provides us with a positive score and negative score for every synonym set (which contains all words which are similar in meaning). Use the first synonym set for the word in SentiWordNet as the first contains the most used words with most common meanings. One could also take an average of the synonym sets. The following logic can be used to represent how positive and negative scores may be set as a weight for synonym set:

```
If pos_score > neg_score, use
pos_score as weight
If pos_score < neg_score, use
(-) neg_score as weight.
```

Feature vector for the above example in SVM would be [0, 0, 0, 1, 0, 0, 0], Pasta is 0 in this case as it is neither positive nor negative and therefore has a subjectivity score 0.

Finally take the majority preference and the percent of tweets with that sentiment and print it as output.

5 Applications and Future Scope

1. Computing customer satisfaction metrics

Sentiment analyzer is exploited in product based companies which rather than manually checking feedbacks can use a sentiment analyzer which could tell whether the product reviews are positive or negative. They can then segregate the positive and negative reviews, focusing on the negative reviews they can try to make their product better.

2. Prediction of elections

The corpus for the reaction of speeches and ideologies of the contesting candidates can be collected and can be categorized as positive or negative collectively or individually, hinting the candidate which the general public is supporting. It could be used by the politicians well to strengthen their campaign by mining the reaction of the masses to his offerings.

3. Stock market

Sentiment about a company could be analyzed from the latest corpus and using that data, trends in the rise and fall of a company's stock can be carried out, mining the news and public reaction to products/services of a company can indicate a rise or fall in the share prices. Sentiment analysis can be exploited well by stock investors.

6 Conclusion

The work presented in this document describes the ways to create a Twitter Sentiment Analyzer. The paper explains the two techniques used for opinion mining. First one is

Rule-based, and the other one is Machine learning based approach. Both the methods have their own advantages and disadvantages. Two of the machine learning based algorithms – Naïve Bayes and SVM - is explained in length. This document annotates the process of opinion mining which has several applications in social media platforms, review related websites or in business intelligence (companies want to know customer's reaction to a product or service).

As a prospective direction for future research and the scope of the analyzer, we intend to use the technique of machine learning for other social media platforms like Facebook. Government officials can also take the benefits of the analyzer for the creation of policies for future.

References

1. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: Proceedings of the 7th International Conference on Language Resources and Evaluation. Valletta, Malta, pp. 1320–1326 (2010)
2. Kumar, A., Dogra, P., Das, V.: Emotion analysis of Twitter using opinion mining. In: 8th International Conference on Contemporary Computing (IC3), Noida, India. IEEE Press (2015)
3. Hasan, K.M.A., Sabuj, M.S., Afrin, Z.: Opinion mining using Naïve Bayes. In: International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Dhaka, Bangladesh. IEEE Press (2015)
4. Bhatia, S., Bhatia, K.K., Sharma, M.: Strategies for opinion mining - a Survey. In: 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India. IEEE Press (2015)
5. Balahadia, F.F., Fernando, Ma, C.G., Juanatas, I.C.: Teacher's performance evaluation tool using opinion mining with sentiment analysis. In: Region 10 Symposium (TENSYP), Bali, Indonesia. IEEE Press (2016)
6. SentiWordNet. <http://sentiwordnet.isti.cnr.it/>
7. University of Toronto, Faculty of Applied Science and Engineering. http://chem-eng.utoronto.ca/~datamining/dmc/support_vector_machine.htm
8. Kumar, A., Sebastian, T.M.: Sentiment analysis of Twitter. *Int. J. Comput. Sci. Issues* **9**(4), 372–378 (2012)
9. Fang, X., Zhan, J.: Sentiment analysis using product review data. *J. Big Data* **2**, 5 (2015). doi: [10.1186/s40537-015-0015-2](https://doi.org/10.1186/s40537-015-0015-2)