ISSN No. 0976-5697

Volume 11, Special Issue I, May 2020

International Journal of Advanced Research in Computer Science

CONFERENCE PAPER

Available Online at www.ijarcs.info

OPINION MINING OF TWITTER DATA USING MACHINE LEARNING

Sailaja Thota, Hanish S P, Raju Y
School of C and IT,REVA University,
Bangalore, India
sailajathota@reva.edu.in, sphanish123@gmail.com, rajuking9056@gmail.com

Abstract — **People express their opinions through social media sites like Twitter, Instagram, and Facebook. Tweets can be used to analyze opinions expressed towards various keywords, which can be anything from the names of products, companies or famous people. Consumers can utilize the insights of analyzed opinions and companies that want to monitor the public sentiment of their brands or any other situation where data on public opinions can be useful. Training data is abundantly available and can be obtained through automated means, we've obtained our data from twitter. The goal of our project is to implement the most effective algorithms and compare their accuracy. We use machine learning techniques like Max entropy, Naive Bayes, SVM and DCNN (Deep Convolution Neural Network). The data is analyzed for the presence of emoticons and keywords that either signify positive or negative expression towards the query term by Max, NB, and SVM methods. In DCNN, latent contextual semantics relationships and co-occurrence of statistical characteristics are used to obtain word embeddings between the words. Finally, word embedding is combined with n-grams and a polarity score is issued for the tweets. Our research focuses on comparing the accuracy of these algorithms to determine which are the most effective and accurate.**

Key Words—Twitter, opinion mining, convolution neural network.

I. INTRODUCTION

Microblogging websites like Twitter have evolved to become a source of varied information. People express their thoughts in short messages known as tweets which are limited to 140 characters. Emotion or sentiment expressed through tweets can be analyzed using machine learning techniques. Beyond that sentimental words can also be analyzed. The sentiment of a tweet can either be classified as positive, negative or neutral. This approach can be used to classify the overall sentiment towards a query term.

Using machine learning we can accurately predict the sentiment of the general population towards various products. Consumers can use sentiment analysis to research about various products that they are either hoping or planning to purchase, by doing so they can make a more conscious decision. Marketers can use this public opinion of their company and products. Generally, tweets are not as thoughtfully composed as reviews, they are more of an informal review or general opinion. By analyzing a large data set of tweets towards a query we can effectively determine the general public opinion expressed towards that query.

II. EXISTING WORK

Most of the studies existing in this matter can be dived into two categories. Supervised methods and lexicon-based methods. Supervised methods are based on training classifiers, like Naive Bayes, SVM, and Max Entropy[2]. These are some of the most commonly used supervised methods in sentiment analysis. These methods use feature extraction through methods like POS (parts of speech tagging), Unigram and Bigram[2, 3] by analyzing contextual information like hashtags, emoticons, capital words, etc. Lexical-based methods determine the overall sentiment of the text by utilizing pre-established lexicons of word weights.

In many cases, the sentiment is associated with the semantics of the context. Supervised learning methods like NB, SVM and Max Entropy don't take in consideration the semantics of the context, by focusing only on the features, the overall sentiment of the text is affected. Deep Convolution Neural Network[1] can overcome this drawback with higher accuracy.

Twitter contains an enormous number of texts. The collected corpus from twitter data will have tweets from different social and interests groups[4].

Research suggests that SVM has the highest accuracy compared to NB and Max Entropy but fails to perform under small observations [9]. DCNN does not face the same issue, it performs even for the smallest tweets.

III. PROPOSED MODEL

A. *Feature representation*

- **Baseline features**

Research suggests that Unigram and Bigram are some of the most common statistical language models used in sentiment analysis that provides state-of-the-art performance for classification on twitter corpus. Hence, we are using Unigram and Bigram features as our baseline features [8].

- **Word sentiment polarity score feature**

The individual polarity score of the words of tweets are summed up to form the total polarity score of the entire tweet. Sentiment score of the words in a tweet is calculated using PMI method which is point-wise mutual method. ***PMI*** is calculated between individual words and the entire tweet. This method helps us identify the sentiment of individual words in the context of the entire tweet which leads to increased accuracy. Following is the formula used for calculating ***PMI***:

$$SenScore(w) = PMI(w, pos) - PMI(w, neg)$$

Here, w is a word in the lexicon, $PMI(w, pos)$ is the PMI score between w and the positive category, and $PMI(w, neg)$ is the PMI score between w and the negative category. A negative $SenScore(w)$ indicates that there is stronger relationship between the word w and negative sentiment and vice versa.

- **Word representation feature**

By using word vector representation feature for tweets, we can capture the grammatical and semantic characteristics of the word. In our study we have implemented ***GloVe*** model for word-level embeddings. ***GloVe*** or Global Vector model is an unsupervised learning algorithm which is used to obtain vector information of words. To understand how GloVe works, lets assume X is a matrix of word-word occurrence count. X_{ij} is the number of times word j occurs in the context of word i. Let, $\sum_k X_{ik}$ be the number of times a word occurs in the context of word i. $P_{ij}=P_{(j|i)}=X_{ij}/X_i$ is the probability that a word j appears in the context of word i.

B. *Deep Convolution Neural Network model*

First, a word vector table is generated using the GloVe model. For a tweet t with m tokens, each token is mapped to the corresponding word vectors by looking up the words vector table which is generated by the GloVe model. V is the vocabulary of words and n is the dimension of the vector. Each word w is mapping to a vector. The tweets are expressed as a vector of the word embeddings.

For the sake of understanding the model let's consider a tweet t with m tokens and a word vector table $L \in R^{(nxV)}$ is generated by the GloVe model. Every word is mapped to a vector $w_i \in R^n$. The tweet is expressed as vector of the word embedding concatenations. Then the features vector from unigram and bigram is concatenated with word embeddings features.

$$v = w_1 + w_2 + w_3 + \ldots + w_{m+2} + w_{m+3} \quad \text{--(1)}$$

Then in the convolution layer, the local sentiment feature vector for each possible word window size is generated. This is generated by employing multiple filters with variable window size h. The bias term and transition matrix are generated for each filter, where h is the number of hidden units in the convolution layer. Each convolution operation generates a new context local feature vector in a word window h.

$$x_i = f(W \cdot v_{i+h-1} + b) \quad \text{--(2)}$$

Here, f is non-linear active function and is the local vector from position i to position $i + h - 1$ in the vector v.

A local feature mapping vector for every word window is generated by the convolution filter, followed by a convolution operation to generate a vector, which can be expressed as

$$x = [\ x_1, x_2, \ldots, x_{n-h+1}\ ] \quad \text{--(3)}$$

K-max pooling is implemented on the new feature vector x generated by the convolution layer. The length of vector depends on how many hidden layers the user wants in the convolution layer. K-max pooling selects the top k number of features. By using this pooling method we can solve the problem that traditional methods can't express negative words.

$$v = max\{ x_1, x_2, \ldots, x_{n-h+1} \} \quad \text{--(4)}$$

The output layer generates probability value of positive or negative sentiment.

$$y^{(j)} = W^{(j)} + y^{(j-1)} + b^{(j)} \quad \text{--(5)}$$

The probability distribution over the sentiment is

$$P(i|\mathrm{t}, \Theta) = \frac{\exp(\mathrm{yi\ j})}{\sum_{k=1}^{n} \exp(\mathrm{yk\ J})} \quad \text{--(6)}$$

IV. EXPERIMENT

A. *Datasets*

The dataset was obtained from twitter using the twitter API. The dataset is an assorted collection of tweets on various topics. The tweets are varied and are not subjective towards a certain topic. By using this method, we have made sure that the tweets cover wide subject matter and aren't too focused around one area.

B. *Baseline*

Two methods have been used for the baseline, Unigram and Bigram. The baseline is used as a point of reference for the DCCN model[11].

C. *Preprocessing*

Preprocessing is done to reduce the noise in a tweet. A tweet will usually have grammatical errors, acronyms, slangs, informal language, etc which can potentially affect the accuracy of the algorithm. Hence, it is important to perform preprocessing to improve the overall accuracy of the algorithm.

D. *Feature Extraction*

We use Unigram and Bigram models for the baseline feature extraction. We issue a word sentiment score for the words in tweets. This lexical based feature extraction method helps us predict the sentiment of a tweet more accurately. ***PMI*** (point-wise mutual information) is used to compute the sentiment score of each word, where the sentiment score of each word is measured between the word and the negative or positive sentiment classification of the entire tweet.

E. *Naïve Bayes*

Naïve Bayes is a probabilistic and supervised learning algorithm that uses the Baye's theorem. The way the algorithm works is by calculating the probability for each tag for a given text. Naïve Bayes theorem states the following theorem, where $\boldsymbol{x_1}$ to $\boldsymbol{x_n}$ are dependent features and $\boldsymbol{y}$ is the class variable:

$$P(y \mid x_1,\ldots,x_n) = (P(y)\ P(x_1,\ldots.,x_n / y))/P(x_1,\ldots.,x_n)$$

Despite being a very basic algorithm, Naïve Bayes continues to be a popular choice for natural language processing tasks and delivers favorable results.

F. *Max Entropy*

Max Entropy is a probability distribution estimation technique. Here the constraints are derived from labeled training data. These constraints are then represented as expected value of the features. The underlining theory of this algorithm is that the distribution should be as uniform as possible.

It is often utilized in natural language processing tasks where the data is not much. By strongly relying on the labeled training data, it provides accuracy that is usually somewhere in the range of 60% to 80%.

G. *SVM*

Support vector machine solves the traditional text categorization problem effectively; generally outperforming Naïve Bayes as it supports the concept of maximum margin. The main principle of SVM is to determine a line which separates all the classes in a search space in such a way the it does so with maximum distance. If represent the tweet using t, the hyper plane using h, and classes using a set into which the tweet has to be we classified, the solution is written as follows equivalent to the sentiment of the tweet.

H. *DCNN:*

Deep Convolution Neural Network employed by us in the project has been discussed in detail in section II under the title 'Proposed Model'.

V. EXPERIMENT RESULTS

The following table is for the accuracy of methods when baseline method was used for preprocessing. It is observed that the actual accuracy from the results is a lot lower than the expected accuracy.

Method	Preprocessing	Expected Accuracy	Actual Accuracy
NVB	Baseline	75%	70%
Max Ent	Baseline	70%	63%
SVM	Baseline	68%	56%
DCNN	Baseline	75%	72%

The follow table is for the accuracy of methods when GloVe model was used for preprocessing. It is observed that in some cases the actual accuracy is higher than the expected accuracy. This showcases the increase in accuracy that GloVe model can have.

Method	Preprocessing	Expected Accuracy	Actual Accuracy
NVB	GloVe	80%	76%
Max Ent	GloVe	70%	73%
SVM	GloVe	80%	81%
DCNN	GloVe	85%	83%

GloVe method leads to better accuracy compared to the baseline method.

VI. CONCLUSION

To determine the most effective way to analyze the opinions collected through Twitter, various machine learning algorithms such as Max entropy, NB, SVM, DCNN were compared for their accuracy. Existing studies were divided into categories such as supervised and lexicon based methods. Preprocessing step is implemented in two ways, baseline and GloVe. DCNN was proved to be showing better accuracy in both baseline and GloVe methods through our experiments. GloVe displayed better accuracy when compared to baseline method.

REFERENCES

[1] Jianqiang, Zhao, Gui Xiaolin, and Zhang Xuejun. "Deep convolution neural networks for twitter sentiment analysis." *IEEE Access* (2018).

[2] Go, Alec, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision." *project report, Stanford* (2009).

[3] Agarwal, Apoorv, et al. "Sentiment analysis of twitter data." (2011).

[4] Pak, Alexander, and Patrick Paroubek. "Twitter as a corpus for sentiment analysis and opinion mining." *LREc*. Vol. 10 (2010).

[5] Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." *Foundations and Trends in Information Retrieval* (2008).

[6] Liu, Bing. "Sentiment analysis and opinion mining." *Synthesis lectures on human language technologies*, University of Illinois at Chicago (2012).

[7] Bhuta, Sagar, et al. "A review of techniques for sentiment analysis Of Twitter data." IEEE, (2014).

[8] Angiani, Giulio, et al. "A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter." (2016).

[9] Desai, Mitali, and Mayuri A. Mehta. "Techniques for sentiment analysis of Twitter data: A comprehensive survey." IEEE (2016).

[10] Jain, Anuja P., and Padma Dandannavar. "Application of machine learning techniques to sentiment analysis." IEEE, (2016).

[11] Severyn, Aliaksei, and Alessandro Moschitti. "Twitter sentiment analysis with deep convolutional neural networks." (2015).