International Journal for Modern Trends in Science and Technology, 6(12): 355-359, 2020

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST061266
Available online at: http://www.ijmtst.com/vol6issue12.html

Twitter Sentimental Analysis

Aditya Prakash

[1]Student, Department of Information Technology, Maharaja Agrasen Institute of Technology, Delhi, India

To Cite this Article
Aditya Prakash, "Twitter Sentimental Analysis", *International Journal for Modern Trends in Science and Technology*, 6(12): 355-359, 2020.

Article Info
Received on 12-November-2020, Revised on 05-December-2020, Accepted on 11-December-2020, Published on 15-December-2020.

ABSTRACT

Twitter sentiment analysis (TSA) provides the methods to survey public emotions about the products or events associated with them. Categorization of opinions through tweets involves a great scope of study and may yield interesting results and insights on public opinion and social behavior towards different events, services, product, geopolitical issues, situations and scenarios that concern mankind at large. These attributes are expressed explicitly through emoticons, exclamation, sentiment words and so on. In this paper, we introduce a word embedding (Word2Vec) technique obtained by unsupervised learning built on large twitter corpora, this process uses co-occurrence statistical characteristics between words in tweets and hidden contextual semantic interrelation

KEYWORDS: *Neural Network, Twitter, CNN, Naive Bayes, SVM, Sentiment Analysis and KNN.*

I. INTRODUCTION

Twitter, with over 145 million daily active users, has now become a potential gold mine for individuals and organizations who have a strong economic, social and political interest in enhancing and maintaining their reputation and clout. Sentiment analysis provides a method of surveying various social media sites of various organizations in real time. For example, users of social media sites mostly represent their views or feelings, and they often share their opinions about daily events or news with the friends or public. Emotional analysis concern mood categories (e.g. joy, angry, satisfaction, happiness) whereas Text sentiment analysis involves categories such as negative, neutral, and positive. In this paper, text sentiment analysis is used to classify the given text into sentiment categories. On social sites, people are likely to post their suggestions or comments that probably contain opinions or sentiment. It will be applicable to various industrial fields if sentiments are accurately predicted. Most of the models of Twitter sentiment analysis based on the method proposed by Pang et al. [10] and build a classifier using machine learning techniques from tweets with manually assigned polarity label. Although the machine learning (ML) methods have shown quite impressive performance, but they strongly depend on manually-defined attributes, which requires more work of domain experts. Due to this, deep learning techniques have been drawing more attention recently, as they may achieve relatively high performance (e.g., accuracy) and reduce the work for the feature definition. In this paper, we built an architecture of the Convolutional Neural Network, which is a type of deep learning model for sentiment classification of text data.

II. METHODOLOGY

Tweet Pre-processing

Noisy and unstructured twitter data can affect the performance and accuracy of the methodology being implemented and to avoid

skewness of data and error, we pre-process the tweets prior to feature selection and reduce the noise in micro-blog text.

3.1.1. Uppercase Identification

Uppercase words refer to shouting online and it is considered quite inappropriate or rude. As a result, they too are an excellent give away of the emotion being displayed through the tweet. It can be used to intensify or diminish the emotion bearing words or the emoticon used. We remove the casing and convert all of them to lower case.

3.1.2. Lower casing

The words need to be in a consistent case and as a result we convert all of them to lower case to remove trouble that comes with irregular casing.

3.1.3. URL extraction

Tweets contain URLs and links to different pages or media thus allowing to share more content in a character limited platform or post. URLs can contain images, gifs, articles and other forms of media content that indicate the emotions however, analysis of URLs will mean that we would have to scavenge through the entire internet, which when translated would ultimately lead to lots of processing time, high data load and slower performance. As a result, all URLs in the training and testing tweets are removed which reduces the size of features.

3.1.4. Anonymity filtering

Hashtags, usernames and any form of identity bearing parts of speech are removed to ensure privacy and anonymity of the user and are replaced to reduce feature size. Hashtags can be extracted and collected separately to classify trending tweets with respect to time and geo-tagged information based on the number or retweets and tweets with that specific hashtag.

3.1.5. Removal of Punctuation

In this step, we remove punctuation from tweets to reduce the noise in tweets.

3.1.6. Removal of Stop words

In this step, we remove words that have little or no significance in the process of sentiment analysis and words which do not add a substantial value to the process and common words such as articles.

3.1.7. Removing Skewness in Dataset

This is done by under-sampling or over-sampling. Over sampling creates a steadier data set by increasing the number of occurrences in the minority class; and under sampling reduces the number of occurrences belonging to the majority class.

Slangs are replaced with their corresponding actual phrase.

Geo-tagged, date-time, number of tweets and re-tweets are removed from data set.

Numerical entries, Duplicate tweets, mentions and Blank spaces are removed.

Replace consecutive non-ASCII characters with a space.

3.1.8. Removal of Emojis

In this step, we remove emojis from tweets to reduce the noise in tweets.

3.1.9. Stemming

Token were reduced to simplest form.

3.1.10. Tokenization

Tokenization describes the general process of splitting the text of a document into a series of tokens in order to identify all words in a given document for further processing, especially to create term document matrix.

3.1.11. Normalization

Data is normalized to be rescaled in unit interval

3.2 Feature Extraction:

2.1. Word Sentiment Polarity Score Feature

TextBlob is a python library used for preprocessing of textual data and facilate us with an API for CNPL task like Sentiment Analysis. We can use TextBlob sentiment analysis feature to obtain polarity score of text. To do so we replace the slangs and abbreviations using a lexicon and the dictionary and all sentiment bearing words along with their corresponding polarity scores, tagging all diminishers and intensifiers with their corresponding sentiment scores so as to give better scores.

The sentiment feature of TextBlob returns two properties subjectivity and polarity. Polarity is float value which lies in the range [-1, 1], where -1 represents negative statement and 1 represents positive statement. Statements which are in range 0 to 1 are considered as positive and statements which are in range 0 to -1 are considered as negative. Negation words which does not belong to any category (positive or negative) are marked zero. Subjective sentences usually refer to judgment, emotion and personal opinion and is a float which lies in the range [0,1].

3.2.2. Word Representation Features

In this step, we used Word2vec model for vector representation of words. Word2Vec is a class of similar models that are used to create word embedding. These are simple two-layer model that are trained to construct semantic contexts of words. This model takes large corpus of text as input and creates a vector space, generally of several hundred dimensions and each unique word being assigned a corresponding vector in the space. Word vectors that share common context in the data set are positioned near to one another in the space.

Consider words βi and βj for which we take βi = gas and βj = solid. The relation between these words can be examined by studying their word cosine distance with various probe words βk. Let Wij be the probability that word j appear in the context of word βi. For words k related to solid but not gas, say βk = solid, we expect the ratio Wjk/Wik will be large. Similarly, for words βk related to gas but not solid, say βk = gas, the ratio should be small. For words βk like effervescent or liquid, that are either related to both gas and solid, or to neither, the ratio should be close to one. Because synonyms and similar paragraphs which have similar context, are mapped to feature vectors that are close to each other. The word vectors produced by word2Vec model can be represented as semantic feature of the tweet.

	1	2	3	4	5	6	7	8	9
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
prince	0	0	0	0	1	0	0	0	0
princess	0	0	0	0	0	1	0	0	0
queen	0	0	0	0	0	0	1	0	0
king	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

Fig 2. Word Representation Features

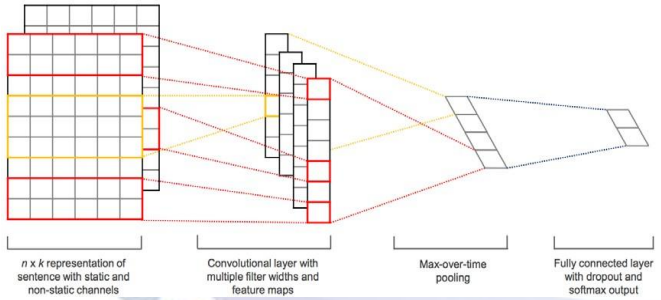

Fig 3: Convolution Neural Network Model

Convolution Neural Network (CNN), which has been widely used on image data base, extracts the important features of the image, as the filter moves through the image. If the input data are given as 1-D, the same CNN function could be used in the text data as well. In the text area, while the "convolutional" filter (i.e., kernel) moves, local information of texts is stored, and important features are extracted. Therefore, using CNN for text classification is effective.

In this paper, embedding layer is obtained through the training process, and all word tokens together with the unknown token for unseen words would be converted to the numeric values using the embedding layer.

3.3 Convolution Neural Network

The A × B matrix, the output of the embedding layer, is then passed as the input in first convolutional layer. The first convolutional layer of filter size C_1 and activation function relu, stores the information needed to categorize the sentiment class in A × B matrix and pass information to the max pooling layer. The C_1 × B matrix convolves all the values of the A × B matrix with a random stride, calculates the dot product, and passes the result to the pooling layer. The second convolutional layer of filter size C_2 and activation function relu, stores the information needed to categorize the sentiment class in A × B matrix and pass information to the next pooling layer. Our model contains four convolution layers with different filter sizes and a pooling layer is added next to every convolution layer, each layer extracts important information stored in the embedding layer. In other words, the

convolutional layers are utilized to extract simple contextual information over the embedding layer output (A × B matrix).

The resultant matrix of the convolutional layer is laid as input to the pooling layer. While L2-norm pooling and average-pooling used as the pooling layer position, in this paper, we used max-pooling. Max pooling is a method for extracting the largest value as a representative of the peripheral values or to extracts the most important features. We adopted max pooling technique to determine the sentiment of sentence as a combination of sentiment of several words rather than expressing sentiment of every word in the sentence. The pooling layer with an arbitrary stride, slides over all the values of the output matrix of the convolutional layer, resulting in output vectors. among several values, it results in much smaller size of output vectors. In this model, we used max pooling layers next to each convolutional layer to shrink the size of output layers. In other words, the convolutional layer extracts the main features of the context, and the pooling layer select the most prominent features.

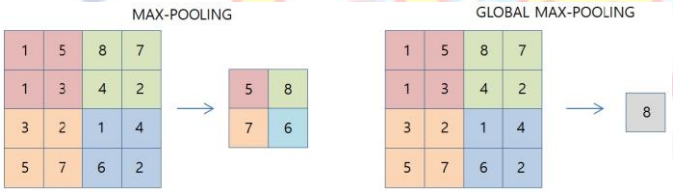

Fig 4: The Difference of max pooling and global pooling.

After passing through the max pooling layer, a concatenate process is performed using 1-D concatenation layer to concatenate output of max pooling layers. Concatenation layer takes list of vectors as input having same shape except for the concatenation axis(axis=1), and outputs a single vector that is the merge of all inputs. We adopted dropout layer to reduce overfitting in model, the idea is to randomly drop some of the neurons while making predictions.

Output of the concatenation layer is laid as the input to the dropout layer with probability of P. We used two dropout layers and after each dropout layer a dense layer is used. In dense layer, each neuron in current layer receives an input from all the neurons present in the previous layer, thus, they are densely connected. In other words, the dense layer is a fully connected layer (i.e. all the neurons present in a current layer are connected to the all the neurons in the next layer). A vector that passes through the dense layer forms an output that is classified as positive or negative. The activation function sigmoid classifies multiple classes in the dense layer. The sigmoid function outputs, the value, which is the probability value, generated for each class.

1)4. Results

The dataset was applied to the proposed CNN model, Naive Bayes model, KNN model, Logistic Regression model and Linear Support Vector Classification model. The experimental results of CNN model is depicted below where each cell correspond to the "+ve" and "-ve" classes, respectively. These tables include the recall, weighted-F1 score, accuracy, F1 score and precision. For example, the F1 scores of the Naïve Bayes is 82.36.

Table 2. The results of models, where Emb (embedding layer), FC(fullyconnected dense layer),GPL(global pooling layer), CL(convolutional layer) and DL(dropout layer) respectively.

Model		
Naïve Bayes	77.13	76.83
Logistic Regression	85.73	84.97
Linear SVC	85.84	82.26
KNN	79.49	80.48
Emb + Conv + gpool + Conv + gpool + FC + Dp + FC + Dp	89.10	87.00
Emb + Conv + gpool + Conv + gpool + Conv + gpool + Conv + gpool + FC + Dp + FC + Dp	90.80	91.88
Emb + Conv + gpool + Conv + gpool + Conv + gpool + Conv + gpool + FC + Dp + FC + Dp	**91.51**	**92.17**

2)4.3 Comparison

In terms of the F1 values, our CNN network was about 8% greater than the traditional machine models. We believe that the main reason of better performance of CNN model is because of its inherent ability to capture higher-level patterns and local patterns through its convolution and pooling layers. The main advantages of convolutional neural network over traditional machine learning methods is to predict the sentiment of a text sentence, we have to look at the sentence as a whole. The polarity of a sentence may be quite varying from the polarity of its words. So, a memory component is required to memorize the past and the future words to predict the probability of the present word. This feature is achieved using neural network not with any other traditional learning model. Also, neural network model is a deep architecture and is able to combine lower-level features to higher-dimension representation while traditional learning methods are non-parametric models and cannot learn highly complex functions. Using these advantages, that word having same sentiment are put together in high dimensional space, neural network is able to achieve high complexity without human intervention.

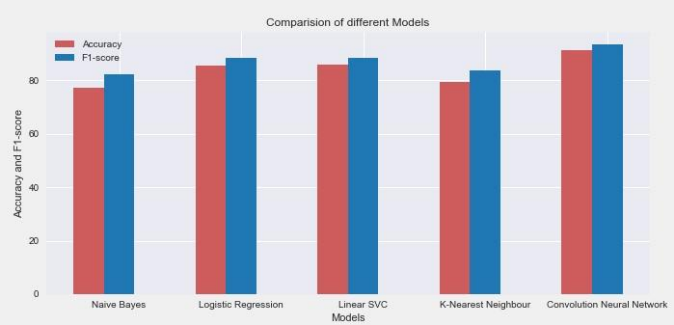

Fig 5: Comparison of Different Models

V.CONCLUSION

In this paper, we proposed a convolutional neural network (CNN) architecture for the binary sentiment classification. By experimental results, we showed that the convolutional layers along with max pooling layers results in better performance. The proposed CNN model achieved an accuracy of 91.51% for the binary classification of text. Interestingly, creating deeper models did not result in regularization techniques such as weight decay and learning rate decay and nor did in terms of model performance. Therefore, we can conclude that convolutional neural networks (CNNs) can be a beneficial tool for the twitter sentiment analysis.

VI. REFERENCES

[1]. Dr.Balika.J.Chelliah, Darshan Lathia, Sandeep Yadav, Meet Trivedi, Shubham Sagar Soni, Sentiment Analysis of Twitter Data using CNN, Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India, Volume 6, Issue 4, April (2018).

[2]. A. Giachanou and F. Crestani, Like it or not: a survey of TSAM.

[3]. N. F. F. Da Silva, E. R. Hruschka and E. R. Hruschka, Tweet sentiment analysis with classifier ensembles, Decis. Support Syst. 66 (2014), 170–179.

[4]. A. Hassan, A. Abbasi and D. Zeng, TSA

[5]. O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada

[6]. N. Kalchbrenner, E. Grefenstette, and P. Blunsom, in Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, vol. 1. Baltimore, MD, USA, Jun. 2014, pp. 655–666.

[7]. ZHAO JIANQIANG, GUI XIAOLIN, AND ZHANG XUEJUN, Deep Convolution Neural Networks for Twitter Sentiment Analysis.

[8]. Ricky Kim. Twitter Sentiment Analysis, Feb 2017. URL: https://github.com/tthustla

[9]. Abhinav Thukral. Sentiment Analysis, July 2017. URL: https://github.com/AbhinavThukral97/SentimentAnalysis