JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Sentiment Analysis of Twitter Data using DL Algorithm and CNN

Anupriya
M.Tech (Artificial Intelligence)
Amity School of Engineering and Technology
Amity University Uttar Pradesh
anupriyamalik0000@gmail.com

Dr. Archana Singh
Professor & Head - Dept of Artificial Intelligence
Amity School of Engineering and Technology
Amity University Uttar Pradesh
asingh27@amity.edu

Mr. Jitendra Singh Jadon
Asst. Professor
Dept. of Artificial Intelligence
Amity School of Engineering and Technology
Amity University Uttar Pradesh
jsjadon@amity.edu

Abstract-Classification of opinions using micro-blogging sites and tweets is a broad area of research. It may yield intriguing results and provide insights into social behavior and public opinion regarding various services, products or events, geopolitical concerns and other scenarios and events that impact the world. In this paper we propose a solution that aims at multidimensional emotion classification that is based on the microblog's emotional classifying Twitter data by using the convolution neural network. In this paper, we will employ n-gram-based features on words using the word sentiment-polarity score feature to create tweets with sentiment features. This will result in a massive amount of data that could be obtained by unsupervised learning. It is intended to be used as a testing set and also a testing set that has cross-validation. Additionally, we will use these features to categorize emotion into five categories: joy, anger, sadness, fear, and surprise. We will also identify data by geotagged information. This feature collection is embedded deep into neural networks using convolution, and its performance is evaluated against other methods like SVM as well as Naive Bayes.

Keywords—CNN, Twitter, Neural Network, Sentiment Analysis, SVM, Naive Bayes

1. INTRODUCTION

Tumblr, Pinterest, and social networking sites like Facebook and Twitter have been potential goldmines for data procurement and purchase. This allows people to study and analyze the social behaviour of others around the world. It can be used to evaluate and predict people's opinions regarding various categories, such as current affairs, geo-political changes and their feelings about specific issues. This includes things as trivial as product reviews or reviews on the future technological advances made by society. This data can be used to predict and analyse state and national security issues and is also used for surveillance. The classification of data from different locations gives a deeper understanding of the emotions and sentiments of people all over the globe. Product owners and companies can use this data analysis to improve their products or services. Consumers can also use this data analysis to get a holistic view of various products, issues and reviews. It helps consumers make informed decisions and align their opinions. These microblogging sites like Twitter produce heterogeneous data. We propose to perform sentiment analysis (SA) on these tweets. Formatting text can be a problem with both informal and formal data. However, Twitter has imposed a limit of 140 characters per tweet. This limit works in our favour and allows us to work with a fixed length. Twitter data corpora can be obtained from several APIs and libraries publicly accessible over the internet. You can preprocess the corpora as well as the raw data that must be classified and processed. We need to preprocess the raw data that is being used for sentiment analysis. This would allow us to extract geo-tags and hashtags from the input data and then replace abbreviations and slang with proper speech. Tweets will replace the emotions that emoticons represent. Tweets will be given a score of +1 for positive, -1 for low-quality tweets, and 0 for neutral. We would also use conjunctions and language intensifiers to improve the scoring system. We will also see the various ML methods that can be used for sentiment analysis and related works as we go along.

2. RELATED WORK

A System to Derive Hidden Affinity Relationships on Twitter Utilizing TextBlob and MongoDB[1] describes how two users can establish relationships using tweet analysis and derive relationship scores using TextBlob or MongoDB. It also uses the REST Twitter API, which is available over the internet. It is limited in the number of tweets it can extract at any given time, preventing real-time analysis of real-time traffic and data. It won't be easy to scale the system and its output to handle larger data sets. This also means that the data sets need to be frequently updated. Redundance could occur in the sense of performing more than two rounds to analyze the previously analyzed data sets. This can lead to a considerable processing time and may cause redundancy. Opinion Mining and Sentiment analysis on a Twitter data stream[2] has done SA using decision trees (SMO), NB classifier, random forest algorithm, and decision trees. The NB classifier was unable to attain the optimum accuracy level.
Additionally, the skewness of the data sets appeared to have a negative impact on recall and affect the accuracy of classifiers. Nonpolar tweets were ignored and treated as such. Further development is possible in areas such as comparison handling and context switching. The Microblogging Sentiment analysis with Lexical Based or Machine Learning Approaches[3] combined the standard ML techniques with lexical-based methods. This seemed redundant, considering ML strategies' higher performance, accuracy rates, and efficiency. SVM, kNN, Maximum

Entropy(ME), and Multinomial Nave Bayes (MNB) were all used in the paper. Lexical-based approaches depend highly on a lexical databank and a language structure that transforms into an opinion classification matrix. ML approaches were more accurate, but they relied on many factors, like feature extraction mechanisms for sentiment analysis. Sentiment Analysis with Sentiment Features[4] uses Sentiment Lexicon to generate a new set of features that will help train SVM (Support Vector Machine), classifier, and outperform the unigram baseline. It tags sentiment-bearing words in a document with the lexicon and assigns them scores. It calculates the features and extracts them through the tagging process. The above study could fluctuate greatly and show deviations from the expected outcome.

Furthermore, the testing and training sets are not identical or close to identical. There is also no new methodology for handling negations. Twitter Data Sentiment Analysis Using Machine

Learning Approaches, Semantic Analysis, and Learning Approaches[5] also perform sentiment analysis using SVM and NB on feature vectors. This is an adjective that has some meaning from the data set. Nave Bayes and the unigram model have produced better results than either alone. The data sets might be less effective for larger sizes. Chinese Microblogging. Emotion Classification using Support Vector Machine[6] uses SVM to classify and distinguish between motions using emoticons. It also has automatic annotations of the corpus that are acquired via APIs and open-source libraries. The methodology has achieved 71% and higher efficiency by combining bigram and unigram features with SVM. This was possible with a chi-square feature selection method and a dataset of 2500 tweets.

Exploring Sentiment Analysis with Twitter Data (3) Like (3) uses domain-independent and domain-specific lexicons to achieve a domain-oriented approach and analyze the sentiments of smartphone users towards brands. This uses NLTK to tokenize tweets and tags them with parts of speech. It does not extract polarity values from lexicon resources. Geotagged features can be used to develop the system further and compare data analysis with market statistics. Efficient Twitter Feed Classification[8] seeks to reduce the data set, feature sets, storage requirements, and computation time necessary to attain a certain accuracy level using one of the machine-learning approaches, such as SVM or Nave Bayes. This is done through the Chisquare feature selection method. These results demonstrate that reducing processing time and input data is possible while still maintaining acceptable accuracy and efficiency. Multi-Lingual Social Sentiment Analysis of Twitter Data by using classification algorithms[9] aims at performing sentiment analysis in multiple languages rather than in English using NB or ME classifiers. They have achieved a 74% accuracy rate. This paper aims to create sentiment analysis in numerous input and output languages to attain geographical diversity. Analysis and Visualization using K-means Clustering [10] uses k-means to extract and process Twitter data and cluster them according to geotagged information using R language with its libraries and functions.

3. PORPOSED MODELLING

3.1 CONVOLUTION NEURAL NETWORK FOR SENTIMENT ANALYSIS

3.1.1. N-grams features

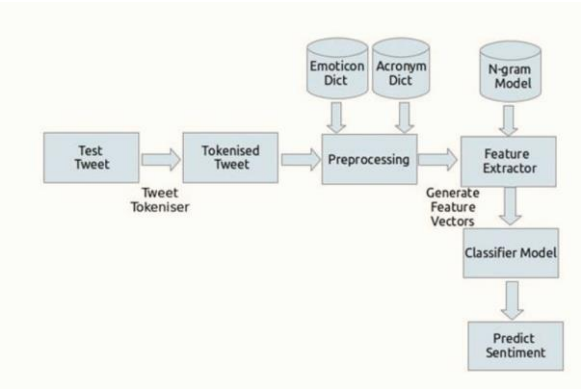

Figure 1. Sentiment analysis workflow

We use the unigram and bigram features of its natural language analysis model as our baseline feature models. This is one of the easiest and most effective natural language models. A unigram refers to an N-gram in length whose TFIDF score (or another important feature) is important. TFIDF score, a weighting system to emphasize words biased towards one of the classes, is often found to perform better than unigrams.

Tf - idf(ωi)=tf(ωi)*log2((N*Pi)/(P*Ni))

3.1.2. Word Sentiment Polarity Score Features

We can use this lexicon-based sentiment polarity feature for tweet sentiment analysis. To extend this feature and get a polarity score, we can use the AFINN dictionary and Senti-WordNet. We replace abbreviations with slang with a dictionary and the Lexon. We tag all sentiment-bearing words with their corresponding sentiment score to get better scores. Negative words can also be tagged. If the words don't belong to any category, they are marked 0. To strengthen sentiment-bearing words, we use intensifiers. The score is annotated to the intensifier. We also use diminishers to reduce the strength of sentiment-bearing words that appear following a diminisher word. Negations can be handled by flipping the score of sentiment-bearing words after a negation. We then weaken the above-flipped polarity by 1. We ignore the 0 tags between the sentiment-bearing words and the valence shifters in the above cases.

3.1.3. Word representation features

Learning from large text corpora with unannotated data can be used to create word representation vectors. Pretrained word embeddings. The Global Vector for word representation is a log bilinear regression modeling that can be used with the global matrix factorization method and local context window. This model trains non-zero elements within a word co-occurrence matrix. Take the words oi, j and consider them as words. Oi is solid, whereas oj is gas. You can examine the relationship between these words by comparing their co-occurrence probabilities to various probe words wk. Let Pij be the probability of word j appearing in context with word oi. We expect that the ratio Pik/Pjk for words k that are solid but not gases will be high. The ratio for words wk that are related to gas but not liquid should be smaller. The ratio should be close at one for words wk such as liquid or effervescent that can either be related to solid or gas or both. Synonyms and similar paragraphs with similar context are mapped to feature-vectors that are close together. The word vectors can then be used as

semantic features in a tweet after they have been trained using the Glove model. You can combine the vectors to create tweet semantic sentiment features.

Figure 2. Word representation features

1.1 TWEET PRE-PROCESSING

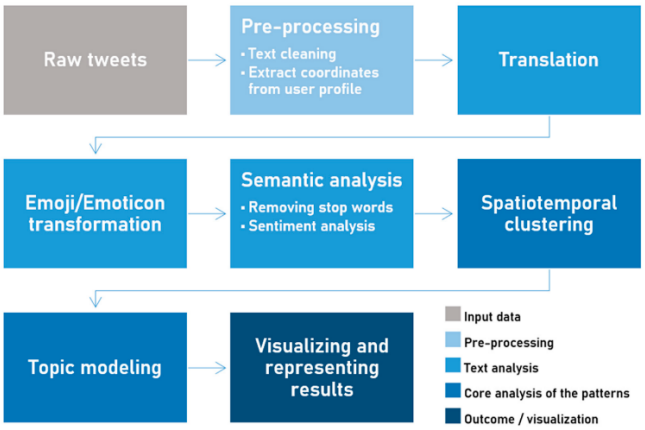

Figure 3. Tweet pre-processing workflow

Unstructured and noisy twitter data can impact the performance and accuracy of the methodology being implemented. To avoid error and skewness, we pre-process the tweets before feature selection. We also reduce noise in micro-blog texts.

3.2.1. Replacing emoticons

Emoticons can be used to express emotion in tweets or messages. They help us distinguish between tweets with sentiments, but no words attached, known as polar tweets. We choose 256 emoticons with the highest frequency of use and most emotional inclination. Then we translate them into five emotions: joy, anger, sadness, fear, worry, surprise, and sorrow. These emoticons can be used to label experimental corpus. The emotional corpus can be automatically labelled as

Emotional type = i category emotion if (Ni < Nj)
j category emotion if (Nj<Ni); for(i! = j)
Not marked if (Ni = Nj)

Where i and j are seven types of emotion: happiness, fondness, microblog's Ni, Nj, and Nj sections contain the emoticons for i and j emotions.

3.2.2. Uppercase Identification

Uppercase words are used to refer to online shouting, and it is considered rude or inappropriate. They are a great way to convey the emotion expressed through the tweets. Using words or emoticons can use them to increase or decrease the feeling. All letters are converted to lowercase by removing the casing.

3.2.3. Lower casing

To avoid confusion caused by inconsistent casing, all words must be written in the same case.

3.2.4. URL Extraction

Tweets can contain URLs as well as links to other pages or media. This allows you to share more content on a limited platform or post. URLs can include articles, images, and other media that express emotion. However, we will need to search the internet for URLs to analyse them. This would result in slow performance, high data loads, and a lot of processing time. All URLs in training tweets have been replaced, which reduces their size.

3.2.5. Anonymity filtering

To ensure anonymity and privacy, usernames and other identity-bearing parts of speech are removed and replaced with more minor features to reduce feature size. You can extract hashtags from tweets to classify them separately according to time and geo-tagged information. This is done based on how many retweets or tweets are associated with the hashtag.

3.2.6. Identification of Punctuation

You can also use punctuation to indicate the polarity of a message. Exclamation marks, for example, are used to emphasize polar messages. It is possible to put the full stop between letters, which can be used as a short form or abbreviation.

3.2.7. Removal of Stop words

We eliminate words that are not relevant to sentiment analysis. Common words, such as articles or words that do not add substantial value or have low IDF values, are also removed.

3.2.8. Removal of Query Term

To ensure that the required tweets are unbiased, they are pulled using a query term.

3.2.9. Compression of Words

Twitter users are informal in communication and use long words to express strong emotions. We reduce words that contain more than three repetitions of the same character/letter during training and evaluation to a sequence of just three characters.

3.2.10. Dataset Skewness Removal

This can be done either by under-sampling or over-sampling. Over-sampling results in a more balanced dataset, as it increases the instances of the minority class. Under sampling reduces the instances of the majority class.

- Slangs (abbreviations) are replaced by their actual phrase equivalents.
- Geo-tagged information, including date-time, number of tweets and the number of re-tweets, can also be obtained and recorded.
- You can remove duplicate tweets, numerical entries, and blank spaces.

3.2.11. Tokenization

Tokenization is the process of breaking down a document's text into a series of tokens to identify each word in the document and allow for further processing, such as creating a term document matrix.

3.2.12. Stemming

This step reduces the words or tokens to their roots.

3.2.13. Normalization

Normalization is necessary for classification. This allows data to be rescaled at the unit interval. Because the largest scale variable will dominate the measure, normalization is essential.

3.3. CONVOLUTION NEURAL NETWORK MODEL

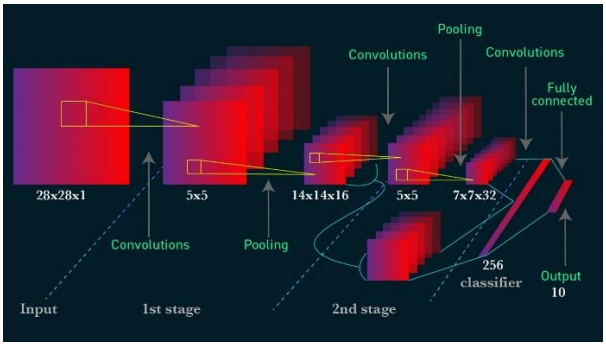

Figure 4. Convolution Neural Network model

Consider a tweet with m tokens. Each token in a Tweet is mapped onto the appropriate word vector by looking at the word vector table

L ϵ R (nx|V|)

V is the word vocabulary, and n is the dimensions of the vector. Each word.

ω i ϵ R n

After mapping, the tweet can be expressed as a vector word embedding concatenation, to which unigram and bigram word sentiment polarity score feature Vectors are applied.

The bigram feature, unigram feature, and on+3 were Twitter-specific features. To unite the representation of tweets with different lengths, the largest of all tweets included in the dataset is considered the default length of tweets in tweet matrixes. For tweets with shorter lengths, a zero vector was added to the rear of the tweet matrix. In the first convolution layer, the convolution calculations are made by employing multiple filters with different window sizes h and generating local feature vector khi for every potential word size window. It is possible to use bias b R as well as a transition matrix of o e (hu*hn) generated for each filter, where hu is the number of hidden units, and hnis the total units of the convolution layer. Every convolution operation generates an additional local feature that is contextual.

χi = f(ω.vi : i + h -1 + b);

The f variable is the non-linear activation function, and the vector v is local, starting at the ith position and ending at (i+h-1)th position in the vector. Convolution filters generate the map of the local features for every possible word-related window within the tweet. This is followed by the convolution process's conclusion to create a brand-new vector.

x={x1,x2,x3,x4…xn-h+1}

Then comes the k-max pooling process that operates on the feature vector x created from the convolution layer. It converts the vector x to a fixed-length vector where length is a hyperparameter that the user determines. It corresponds with the number of layers hidden in the convolution layer. The top k features are chosen using the k-max pooling method akin to the many hidden layers to preserve the sentiment's crucial features. To get better features, we transferred the fixed length vectors generated by the process of k-max pooling into convolution layers to create another vector. Within the modelling, we pick the hidden layers, which include three convolution layers as well as three pooling layers k-max. The convolution layer is composed of three filters, namely t and f., which result in feature maps.

M ϵ R (f*(n-2))

The max-pooling-over-time layer is responsible for selecting the most relevant features within the temporal dimension using lters of size f*(n2). In twitter classification, the resultant outcomes classes can have two polarities, positive and negative, which can be configured using a softmax output with two neurons. The output layer is a softmax layer which generates a probability value for positive or negative sentiment. To adjust the sentiment characteristics of an input layer, the output layer uses a fully connected layer of softmax to generate a probability distribution of the sentiment classification labels

yj = ωjyj-1 + bj ;

yj: output vector of softmax layer.

yj-1: output vector of pooling layer.

wj: transition matrix of softmax layer.

bj: bias factor of softmax layer.

The probability distribution over the sentiment labels is:

$P(i|t,\theta) = ((exp(yi\hat{}j))/\sum(exp(ykj))$

The summation series is defined as k, where k can be anywhere from 1 to N. Dropout regularization is applied to fully connected layers in order to remove the problem of many hidden units and the connections among them.

3.4. EXPERIMENTAL SETUP

We apply a 10-fold cross-validation to each dataset in this paper. The same preprocessing steps were used for all datasets. Each experiment was run with the same preprocessing steps. We trained the convolution neural networks on the training set. The highest accuracy points were obtained in the verification set. We also reported the accuracy of each test set. Each dataset was cross-validated 100 times. This meant that each replication was a 10-fold cross-validation. We compared the performance of each replication to determine the average accuracy and reported the results. We first classify tweets based on their sentiment score. Then we can categorize them as polar or non-polar and positive or negative. The final score and the equivalent word vector representation can be used to assign the tweet to one of these groups: happy, anger or sorrow, fear, surprise, or even fear. We can establish a relationship between two people by counting the number of tweets they exchange. For this scenario, we must avoid using power user tweets such as celebrities or famous figures as input. A higher score indicates a stronger relationship. A lower score means a weaker relationship. The rst will use this value to calculate the overall score. A few overwhelmingly positive tweets will still be more than a longer-lasting, positive relationship.

$F = N \times \sum(Posi - Negi)$

Where F is the final friend value, and N is the total number of tweets a user sends to another. Posi and Negi indicate the positivity or negativity scores for each tweet. Clustering algorithms can be used to group tweets based on the geo-tagged information and the date-time stamp. They can be clustered with hashtags, which will highlight the most current and recent trends on social media and online platforms in the year. This is possible with k-means Clustering. The number of clusters in the K-means clustering algorithm is predefined. They can be randomly selected or created from the data using the Elbow method. The elbow method works by running the k-means clustering algorithm for a set of values on the dataset and calculating the sum of squared errors (SSE) for each value. Next, plot a line chart of each SSE value. The best chart will look like an arm.

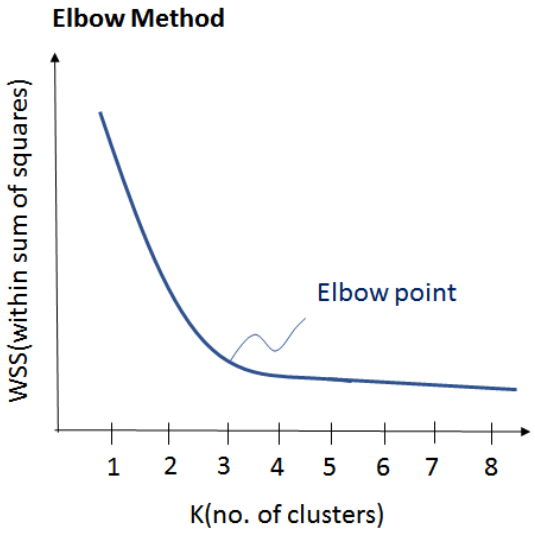

Figure 5. K-clustering graph

All the above experiments will be performed in Python. We will use a Twitter corpus of 25000 tweets to help us. We will also make use of NLTK and Python libraries, lexicons like SentiWordNet, and NLTK libraries and python dictionaries to aid our experiment. This will ultimately yield the result of using convolutional neural networks for sentiment analysis instead of native Machine Learning methods such as SVM and Maximum Entropy, Nave Bayes and K-NN algorithms.

4. DISCUSSION

We find that the Glove model with CNN is the most efficient and accurate at a shocking count of 87%. Using the same SVM approach or BoW, unigram or bigram features to analyze large data sets produces an accuracy rate of 71%. Despite the fact that the CNN model takes a long time to process, it is efficient.

5. CONCLUSION

This paper presents sentiment analysis of Twitter data using convolution neural network algorithms. Instead of machine learning methods such as SVM or Nave Bayes, and we use the global vector representation model to classify emotions into five distinct types. We also propose a system that can establish relationships between users, rank the most important topics on social media and microblogging platforms based on the number of tweets and the number of re-tweets received and group them according to their geographic location. Through the use of REST API, we have attempted to provide a complete analysis of Twitter data. However, there is still scope for further development in sentiment analysis and lexical analyses in areas such as rhetoric tweets or sarcasm. We will continue to explore and develop this study.

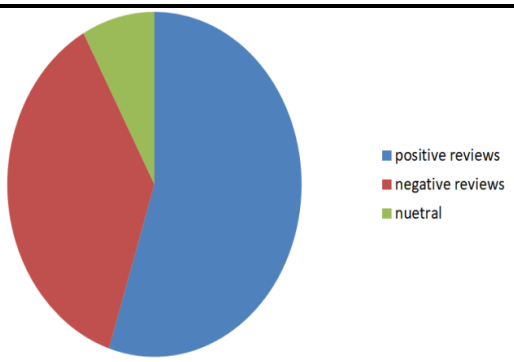

Figure 6. Pie chart showing review distribution

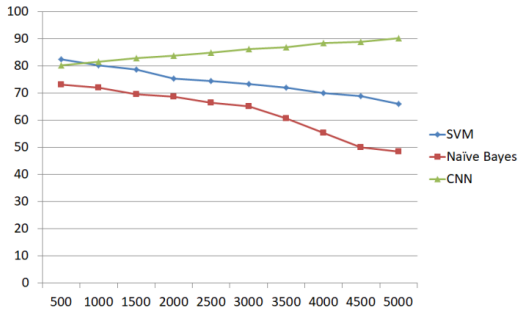

Figure 7. Comparison of algorithms

References

[1.] Opinion Mining and Sentiment Analysis on a Twitter Data Stream Balakrishnan Gokulakrishnan * 1, Pavalanathan Priyanthan *2, Thiruchittampalam Ragavan*3, Nadarajah Prasath*4, A Shehan Perera*5 (Department of Computer Science and Engineering, University of Moratuwa, Moratuwa, Sri Lanka).

[2.] Sentiment Analysis using Sentiment Features Seyed-Ali Bahraini an, Andreas Dengel (Computer Science Dept., University Of Kaiserslautern, Germany Knowledge Management Dept., DFKI, Kaiserslautern, Germany)

[3.] Chinese Microblogging Emotion Classification based on Support Vector Machine Xiao SUN, Changsheng LI, Jiaqi YE Anhui Province Key Laboratory of Affective Computing and Advanced Intelligent Machine (School of Computer and Information, Hefei University of Technology, Hefei, China 230009).

[4.] Microblogging Sentiment Analysis with Lexical Based and Machine Learning Approaches Warih Maharani (Faculty of Informatics Telkom Institute of Technology Bandung Indonesia).

[5.] Optimizing Support Vector Machine in Classifying Sentiments on Product Brands from Twitter Jao Allen Banados,Kurt Junshean Espinosa (Department of Computer Science and Engineering, University of the Philippines Cebu Lahug, Cebu City, Philippines).

[6.] Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis Geetika Gautam, Divakar Yadav (Department of Computer Science and Engineering, Jaypee Institute of Information technology Noida, India).

[7.] Exploring Sentiment Analysis on Twitter Data Manju Venugopalan, Deepa Gupta (Department of Computer Science, Department of Mathematics, Amrita School of Engineering Amrita Vishwa Vidyapeetham Bangalore Campus, India).

[8.] Effect of Training Set Size on SVM and Nave Bayes for Twitter Sentiment Analysis Omar Abdelwahab1, Mohamed Bahgat2 , Christopher J. Lowrance1, Adel Elmaghraby1 (Department of Computer Science and Engineering, University of Louisville, Louisville, KY, USA .

[9.] Efficient Sentiment Classification of Twitter Feeds Nicholas Chaman Singh and Patrick Hosein (Department of Computer Science The University of the West Indies, St. Augustine, Trinidad).

[10.] Collective Intelligence Sentimental Analysis of Twitter Data By Using Standford NLP Libraries with Software as a Service (SaaS) Hase Sudeep Kisan, Hase Anand Kisan ,Aher Priyanka Suresh (Department of IT ,Department of Computer Science and Engineering, Department of Computer Engg,AVCOE, Sangamner, M.S. (India) ,PREC, Loni, M.S. (India) SVIT, Nashik, M.S. (India)).

[11.] A Character-based Convolutional Neural Network for Language Agnostic Twitter Sentiment Analysis Jonatas Wehrmann , Willian Becker, Henry E. L. Cagnini, and Rodrigo C. Barros (Faculdade de Informatica Pontificia Universidad Cattolica do Rio Grande do Sul Av. Piranga, 6681, 90619-900, Porto Alegre, RS, Brazil).

[12.] Sentiment Analysis Based Product Rating Using Textual Reviews Sindhu C, Dyawanapally Veda Vyas, Kommareddy Pradyoth (Department of Computer Science and Engineering, SRM University, Kattankulathur, Chennai, India.).

[13.] Deep Convolution Neural Networks for Twitter Sentiment Analysis Zhao Jianqiang1),2) , Gui Xiaolin1),2)*(1.School of Electronic and Information Engineering, Xian Jiao tong University 2. Key lab of Computer Network of Shaanxi Province P. R. China).

[14.] Analysis and Visualization of Twitter Data using k-means Clustering Neha Garg ,Rinkle Rani (Department of Computer Science and Engineering Thapar University, Patiala, India)