Sentiment Analysis of Twitter Data using CNN

Dr.Balika.J.Chelliah [1], Darshan Lathia [2], Sandeep Yadav [3], Meet Trivedi [4], Shubham Sagar Soni [5]
[1-5] Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.

Abstract – Classification of opinions through tweets and other micro-blogging sites entails a great scope of study and can yield interesting outcomes and insights on social behaviour and public opinion towards different products, services, events, geopolitical issues and situations and scenarios that affect mankind at large. Through this paper, we propose a multidimensional sentiment classification method based on micro-blog emotion classification of twitter data through the use of Convolution Neural Networks. In this paper we will be using n-gram features on words with word-sentiment polarity score feature to form a set of tweets with sentiment feature where a large corpora of data would be obtained through unsupervised learning and this would be utilised as training set and testing set using cross-validation. Moreover, we shall be using these features to classify the emotion in five categories such as happiness, anger, sorrow, fear and surprise and differentiate data through geotagged information. The feature set is integrated deeply into convolution neural networks and its performance is compared with other methods such as SVM and Naive Bayes.

Index Terms—CNN, Twitter, Neural Network, Sentiment Analysis, SVM, Naive Bayes

1. INTRODUCTION

Social networking sites such as facebook, twitter, Tumblr, Pinterest etc have become potential gold mines in terms of the data availability and procurement to conduct and generate all kinds of study and analysis on social behaviour portrayed by people across the globe. It can be potentially used to predict and evaluate peoples opinion regarding varied categories such as the changing geo-political landscape, current affairs and their feelings regarding certain issues that affect the masses such as climate change or something as trivial as product reviews and reviews upon the upcoming technological strides taken by the society at large. Analysis of this very data can be used by governments to study people sentiments and predict issues of state or national security and can also be used as surveillance. Moreover, classification of information from different location provides a deeper insight to the emotions and sentiments of different people from all across the world. Companies and product owners who seek to ameliorate their products and services can use this data analysis for further development. In fact this same data can be used by consumers alike to gain a holistic perspective regarding various issues, products and reviews and helps them make an informed decision and align their opinions accordingly. Data generated by these microblogging sites such as twitter is heterogeneous and we in this paper propose to run sentiment analysis (SA) on the tweets. Usually size of text is an issue with informal and formal data, but thanks to twitter the size limit is of 280 characters per tweet instead of the traditional 140 and this size limit does play to our advantage and help us work with a fixed length. The twitter data corpora will be obtained from various APIs and libraries publicly available over the internet. The corpora can be pre-processed as well as raw data that needs to be classified and processed. To achieve sentiment analysis on this data we would need to pre-process the raw data being taken as input where we would extract geo-tags, hashtags, user names and replace slangs and abbreviations with proper speech and elaborate them. Emoticons will also be replaced by the emotions they signify and tweets would be scored +1 for positive tweets, -1 for negative tweets and 0 for neutral tweets. To refine the scoring system, we would also record and use language intensifiers and conjunctions to better analyze and encapsulate comparative and superlative degrees while running SA on the input tweets. Through the course of this paper we shall also be witness to the other ML approaches that are utilised for sentiment analysis and the related works that have been performed uptill now.

2. RELATED WORK

A System for Deriving Hidden Afnity Relationships on Twitter Utilizing Sentiment Analysis (1) discusses the establishment of relationships between two twitter users through the analysis of tweets and deriving relationship scores using TextBlob and MongoDB and uses REST twitter API that is obtained over the internet. However, it faces the limitation of the number of tweets it can extract at a certain interval thus preventing the analysis of the same in real-time data and traffic. Scaling the system and the output to accommodate larger data and build relationships for all of them is an impending challenge and since the data sets would need to be regularly updated, redundance might occur, in terms of performing more than two rounds of analysis for the previously analyzed data sets and might use up a lot of processing time.

Opinion Mining and Sentiment Analysis on a Twitter Data Stream (2) has performed SA using decision trees, SMO, NB classifier and random forest algorithm. It was found that NB classifier failed to achieve the optimum accuracy level and that skewness of data sets seemed to be a problem to achieve recall and affected the accuracy of the classifiers. Many non-polar tweets were considered irrelevant and calculated

separately. Comparison handling and context switches are also some of the areas wherein further development can be made.

Microblogging Sentiment Analysis with Lexical Based and Machine Learning Approaches (3) used the standard ML techniques along with lexical based approaches which seemed rather redundant given the greater performance, accuracy rates and efficiency of ML approaches. The paper deployed algorithms such as SVM,k-NN, Maimum Entropy(ME) and Multinomial Nave Bayes(MNB) wherein lexical based approach were highly dependent on a lexical database and an established language architecture which transformed into opinion classification matrix. In spite of better accuracy, ML approaches depended on various factors such as feature extraction mechanisms for the sentiment analysis.

Sentiment Analysis using Sentiment Features (4) uses Sentiment lexicon to generate a new set of features that would help train SVM (Support Vector Machine) classifier and outperform unigram baseline. It uses lexicon to tag sentiment bearing words in a document and tag them with their corresponding scores. It computes the features and extracts them by aggregating the words score through the tagging process. However, the above study would fluctuate dramatically and show deviation from the expected outcome and accuracy of the training sets and testing sets are not the same or closely identical and there is no novel methodology in place to handle negations. Sentiment Analysis of Twitter Data Using Machine

Learning Approaches and Semantic Analysis (5) too like the above one performs sentiment analysis using SVM, NB and Maximum Entropy on feature vector which is an adjective from the data set that has some meaning given to it. Using Nave Bayes with the unigram model has given better results than using it alone. However, the data sets may not perform great for a larger size. Chinese Microblogging.

Emotion Classification based on Support Vector Machine (6) makes use of SVM to classify and differentiate between motions through the sole use of emoticons and has automatic annotation of corpus which is acquired through APIs and open source libraries. Using unigram and bigram features with SVM has allowed the methodology to achieve 71 percent and greater efficiency with a chi-square feature selection method and a data set of 2500 tweets.

Exploring Sentiment Analysis on Twitter Data (7) like (3) uses a domain independent and domain specific lexicons to obtain a domain oriented approach and analyse and extract the sentiments of consumers towards smartphone brands. It utilises NLTK to tokenize tweets and tag them using parts of speech instead of extracting polarity values from generic lexicon resources. Further development can be made by use geo-tagged features and comparing the analysis of data with actual market statistics.

Efficient Sentiment Classification of Twitter Feeds (8) aims to achieve the smallest data set, feature set, storage requirements and computation time required to achieve a given accuracy level on one of the machine learning approaches utilised such as SVM, Nave Bayes and Maximum Entropy, through Chi-square feature selection method. The results show that a significant reduction in processing and input data can be achieved while maintaining the accepted accuracy and efficiency levels.

Multi-Lingual Sentiment Analysis of twitter data by using classification algorithms (9) aims to conduct sentiment analysis in more than one languages rather than perform the process in a single language such as English using NB and ME classifiers and have achieved an accuracy rate of 74 percent. The paper seeks to establish sentiment analysis in more than input languages and more than one output languages to achieve geographical diversity of the twitter data.

Analysis and Visualization of Twitter Data using kmeans Clustering (10) uses k-means clustering to extract and process twitter data and moreover, cluster them according to geo-tagged information using R language and its libraries and functions.

3. PORPOSED MODELLING

3.1 CONVOLUTION NEURAL NETWORK FOR SENTIMENT ANALYSIS

3.1.1 N-grams features

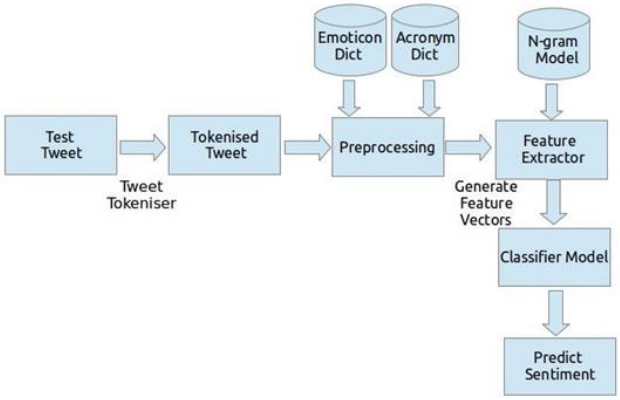

Figure 1. Sentiment analysis workflow

One of the most simple and effective natural language analysis models we use its unigram and bigram features as baseline feature models. A unigram is a N-gram of length on whose TFIDF score is an important feature. The TFIDF score

is a weighting mechanism to emphasize the words biased to one of the two classes and hence is usually found to perform better than the presence or absence of unigrams.

$Tf - idf(\omega_i)=tf(\omega_i)*log_2((N*P_i)/(P*N_i))$

3.1.2. Word Sentiment Polarity Score Features

This is a lexicon based sentiment polarity feature that we can utilise for tweet sentiment analysis. We can use the AFINN lexicon and use Senti-WordNet to extend this feature and obtain a polarity score. To do so we replace the abbreviations and slangs using a dictionary and the lexicon and tag all sentiment bearing words with their corresponding sentiment scores alongwith tagging all intensifiers and diminishers with the corresponding scores so as to provide better scores. Negation words are also tagged and if the words do not belong to any category, they are marked 0. We use intensifiers to strengthen sentiment-bearing words that appear after an intensifier by the score annotated to the intensifier word. Similarly, we use the diminishers to weaken the weaken the strength of sentiment bearing words that appear after a diminisher word by strength of the diminisher. We can also handle negations by flipping the polarity score of the sentiment bearing words that appear after a negation, which we later weaken the above flipped polarity score by 1. In the above cases we ignore the 0 tag that appear between the sentiment bearing word and the valence shifters.

3.1.3. Word representation features

Word representation vectors can also be used by learning from a large text corpora of unannotated data. Using pre-trained word embeddings. We can utilise the Global Vector for word representation model which is a log bilinear regression model that has the advantages of local context window and global matrix factorization method. The model trains non-zero elements in a wor to word co-occurence matrix. .

Consider words ω_i and ω_j for which we take ω_i = solid and ω_j = gas. The relationship of these words can be examined by studying the ratio of their co-occurrence probabilities with various probe words wk. Let P_{ij} be the probability that word j appear in the context of word ω_i. For words k related to solid but not gas, say wk = solid, we expect the ratio P_{ik}/P_{jk} will be large. Similarly, for words wk related to gas but not solid, say ω_k = gas, the ratio should be small. For words wk like liquid or effervescent, that are either related to both solid and gas, 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. After training by the Glove model, the word vectors can be represented as semantic features of the tweet. The vectors can be concatenated as tweet semantic sentiment features.

	I	love	Programming	Math	tolerate	Biology	.
I	0	2	0	0	1	0	2
love	2	0	1	1	0	0	0
Programming	0	1	0	0	0	0	1
Math	0	1	0	0	0	0	1
tolerate	1	0	0	0	0	1	0
Biology	0	0	0	0	1	0	1
.	1	0	1	1	0	1	0

Figure 2. Word representation features

3.2 TWEET PRE-PROCESSING

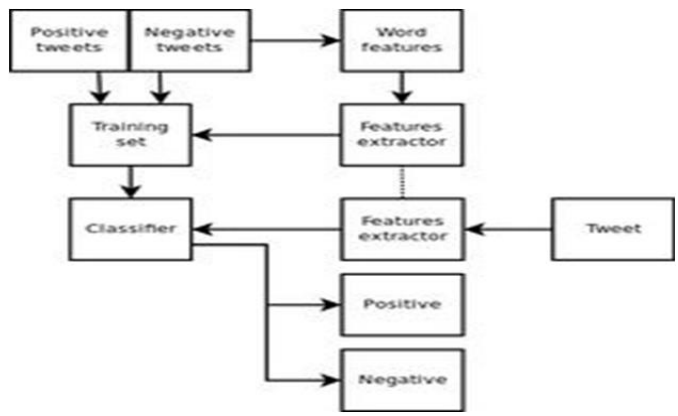

Figure 3. Tweet pre-processing workflow

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.

3.2.1. Replacing emoticons

Emoticons are symbols that are used instead of words to signify emotion embedded in a tweet or a message. They also help us differentiate between tweets that bear emotions without the sentiment bearing words known as polar tweets and those that dont have emotions attached to the tweet such as non-polar tweet. We select 256 emoticon symbols with most distinct emotional inclination and most usage frequency and reflect them into five kinds of emotions: happiness, anger, sorrow, fear, surprise. We can also use these emoticons to voluntarily label experimental corpus. We can automatically label the emotional corpus as

Emotional type = i category emotion *if* $(N_i < N_j)$

j category emotion *if* $(N_j<N_i)$; *for*$(i! = j)$

Not marked *if* $(N_i = N_j)$

Where i, j represent seven emotional types: happiness, fondness, surprise, anxiousness, sorrow, anger and

detestation. N_i , N_j respectively represents a piece of Micro-blog which contains the number of emoticons of i emotion and j emotion.

3.2.2. 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.

3.2.3. 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.

3.2.4. 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 articles, images and other forms of media that signify the emotion 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 tweets are replaces by <url> which reduces the size of features.

3.2.5. 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.

3.2.6. Identification of Punctuation

The punctuations can also give insight to the polarity of the message. For example, exclamation marks are used to express powerful emphasis which are usually polar messages. Putting the full stop between some letters can justify it as being a means of an abbreviation or short form.

3.2.7. Removal of Stop words

We remove words that bear little or no significance to the process of sentiment analysis and common words such as articles and words which do not add a substantial value to the process or do not have a high IDF value.

3.2.8. Removal of Query Term

The required tweets are pulled from using a query term so that the set is unbiased and not bending to a certain emotion or decision prior to the process of prediction.

3.2.9. Compression of Words

Twitter users tend to be very informal in their language and most of them elongate words to express strong emotions. During training and evaluation, for words containing more than 3 subsequent occurrence of the same repeated character/ letter, we reduce it to a sequence of three characters.

3.2.10. Removing Skewness in Dataset

This is done by over-sampling or undersampling. Over sampling creates a more balanced dataset by increasing the number of instances in the minority class; and under sampling on the other hand reduces the number of instances belonging to the majority class.

- Slangs (abbreviations) are replaced with their actual phrase equivalences.
- Geo-tagged information along with date-time and number of tweets and re-tweets is also obtained and recorded.
- Blank spaces, numerical entries and duplicate tweets are removed.

3.2.11. 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.

3.2.12. Stemming

In this step, the tokens or words were reduced to their root form.

3.2.13. Normalization

Data is normalized for classication in order for it to be rescaled to the unit interval. Normalisation is important because without it the measure will be dominated by the largest scale variable.

3.3. CONVOLUTION NEURAL NETWORK MODEL

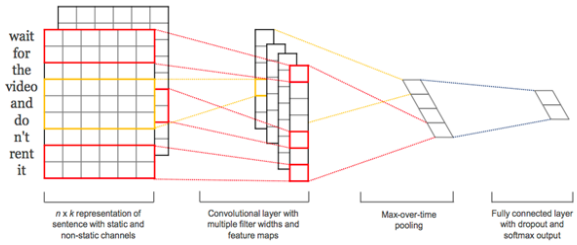

Figure 4. Convolution Neural Network model

Let us consider a tweet with m tokens where each token in a tweet is mapped onto the corresponding word vector by looking up the word vector table

$$L \in R^{(nx|V|)}$$

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

$$\omega_i \in R^n$$

After mapping, the tweet is expressed as a vector of word embeddings concatenation to which unigram, bigram, word sentiment polarity score feature vectors are applied as a feature vector v of tweet

$$\upsilon = \omega_1 \oplus \omega_2 \oplus \omega_3 \oplus \omega_4 \oplus .... \oplus \omega_{n+1} \oplus \omega_{n+2} \oplus \omega_{n+3}$$

$\oplus$ is the concatenation operator of vector.

$$\omega_{n+1} \in R$$

was word sentiment polarity feature vector,

$$\omega_{n+2} \in \{0,1\}$$

was unigram and bigram feature and ω_{n+3} were the twitter specific features. To unify the matrix representation of tweets in different length, the maximum length of all tweets in the dataset is used as the fixed length for tweet matrices. For shorter tweets, zero vector was padded at the back of a tweet matrix. In the first convolution layer, convolution calculation are performed using employ multiple filters with variable window size h, and generate local sentiment feature vector χ_i for each possible word window size. We can use a bias $b \in R$ and transition matrix $\omega \ \varepsilon \ R^{(hu*hn)}$ generated for each filter,where h_u: amount of hidden units and h_n: total units in the convolution layer where each convolution operation will generate a new contextual local feature

$$\chi_i = f(\omega.v_i : i + h -1 + b);$$

where f is the non-linear active function and v is the local vector from ith position to $(i+h-1)th$ position in vector v. The convolution filter generates a local feature mapping vector for each possible word window in the tweet, which is followed by the completion of the convolution operation to generate a new vector

$$x=\{x_1,x_2,x_3,x_4 \ldots x_{n-h+1}\}$$

This is followed by k-max pooling operation that is employed over the new feature vector x generated by the convolution layer. It maps the vector x to a fixed length vector where the length is a hyperparameter determined by the user and corresponds to the number of hidden layers within the convolution network. The top k features are selected through the k-max pooling technique which correspond to the multiple hidden layers so as to retain the important sentiment feature information. In order to obtain better feature information, we fed the fixed length vectors created by the k-max pooling to a convolution layer for obtaining a new vector again. In the model, we select the hidden layers to contain three convolution layers and three k-max pooling layers.

The convolution layer involves τ with f filters $3*d$ resulting in feature maps

$$M \in R^{(f*(n-2))}$$

The max-pooling-over-time layer is responsible for selecting the most relevant features within the temporal dimension by 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 of the architecture is a softmax layer that generates probability value of positive or negative sentiment. The output layer uses a fully connected softmax layer to adjust the sentiment characteristics of the input layer, and gives a probability distribution of the sentiment classification labels

$$y_j = \omega_j y_{j-1} + b_j ;$$

y_j: output vector of softmax layer.

y_{j-1}: output vector of pooling layer.

w_j: transition matrix of softmax layer.

b_j: bias factor of softmax layer.

The probability distribution over the sentiment labels is:

$$P(i|t,\Theta) = ((exp(yi\hat{}j))/\sum(exp(yk^j))$$

Where k runs from 1 to n and so does the summation series. We apply dropout regularization to the fully connected layers to eliminate the problem of a lot of hidden units and the connections between them.

3.4. EXPERIMENTAL SETUP

In this paper, we apply 10-fold cross validation for each dataset. For all datasets, the same preprocessing steps was applied. For each experiment, we trained the convolution neural network on the training set, and obtained the highest accuracy points in the verification set, and reported the accuracy of the test set. We replicated cross validation experiments 100 times for each dataset, so that each replication was a cross validation of 10-fold. We recorded the average performance for each replication and report the mean average accuracy values observed over 100 replications of cross validation. Based on the sentiment score of the tweet we first categorize them into polar and non-polar and then into positive and negative. On the basis of the final score and its equivalent word vector representation we can assign the tweet to any one of the following groups: happy, anger, sorrow, fear

or surprise. The total number of tweets exchanged between two users will help us establish the relationship between them. However for this particular scenario of relationship establishment we have to avoid taking power user tweets as input such as celebrities and famous figures. A higher value suggests a stronger relationship, while a low value suggests a weak relationship. This value will be used to scale the overall score computed by the rst metric. Thus, a small number of overwhelmingly positive tweets will still register less than a long-lasting relationship that is slightly less positive.

$$F = N \times \sum_{i=1}^{N} (Pos_i - Neg_i)$$

where: F is the final friendship value and N is the total number of tweets sent from a specific user to another. Pos_i ; Neg_i are the positivity and negativity scores for the $_{ith}$ tweet. Using clustering algorithms, the tweets can further be clustered according to the geo-tagged information present along with the date-time stamp. Moreover, they can also be clustered using hashtags thus evidently highlighting the most recent and current trends in the year on online platforms and social media. This can be done using k-means clustering. In k-means clustering algorithm, the number of clusters is predefined and selected randomly or generated from the data set using Elbow method. The idea behind the elbow method is to run the k-means clustering algorithm on the dataset for a range of values and calculate the sum of squared error (SSE) for each value. Then plot a line chart for each value of the SSE. If the chart looks like an arm, then the value of the elbow of the arm is the best.

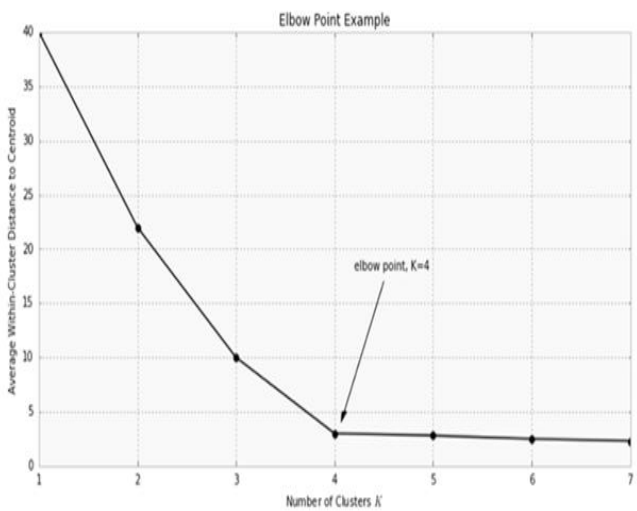

Figure 5. K-clustering graph

All of the above experiment will be done in Python where we will consider a twitter data corpora of 25000 raw tweets that will be pre-processed by the system and we shall make use of NLTK libraries and python libraries along with lexicons such as Senti-WordNet to aid our experiment which will finally yield us the result of using convolution neural networks for sentiment analysis instead of native Machine Learning approaches such as SVM, Nave Bayes, Maximum Entropy and K-NN algorithms.

4. DISCUSSION

At the end we discover that Glove model with CNN produces the highest efficiency and accuracy at a staggering count of 87% whereas using the same on SVM or using SVM approach with BoW or unigram and bigram features for a data set this large produces an accuracy rate of at most 71% and not more. In spite of the processing time taken by the CNN model, its efficiency score is on point.

5. CONCLUSION

In this paper we have conducted sentiment analysis of twitter data using convolution neural network algorithms instead of the machine learning approaches such as SVM and Nave Bayes, by using the global vector representation model and have classified the emotion into five distinct types. Along with this, we have also proposed a system to establish relationships between users and rank the most popular topics in social media and micro-blogging platform by the sheer number of tweets and re-tweets on the basis of the data provided and cluster them according to their location. We have tried to provide complete analysis of twitter data through the usage of REST API. There is however, scope for further development regarding sentiment analysis and lexical analysis in areas of rhetoric tweets and sarcasm, which we shall further explore and develop on the basis of the current study conducted.

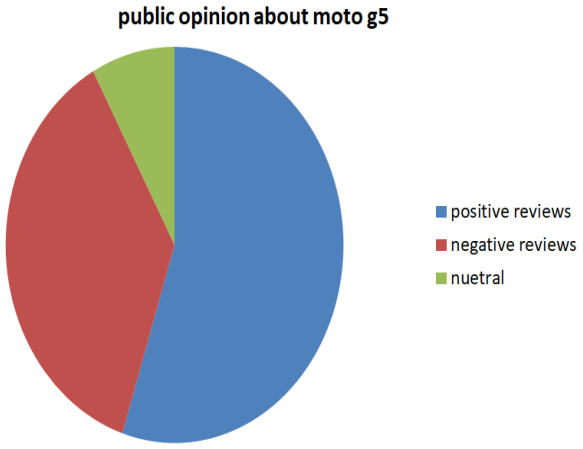

Figure 6. Pie chart showing review distribution

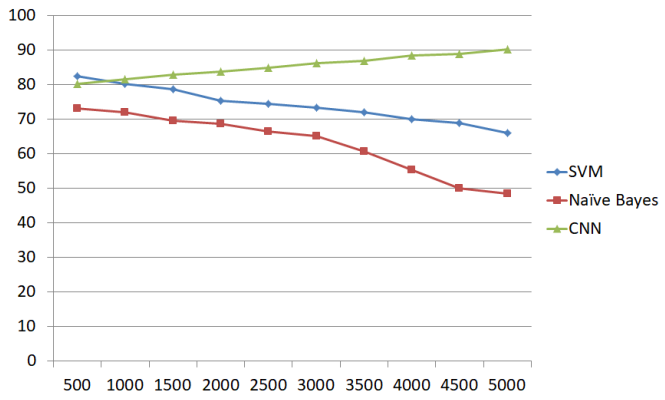

Figure 7. Comparison of algorithms

REFERENCES

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

[2] Sentiment Analysis using Sentiment Features Seyed-Ali Bahrainian, 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, Chengcheng 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 Chamansingh 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 StandfordNLP 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 Pontifcia Universidade Catolica do Rio Grande do Sul Av. Ipiranga, 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 Jiaotong 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).