International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-6, Issue-12)

Research Article | **December 2017**

A Twitter Sentiment Analysis Using NLTK and Machine Learning Techniques

[1]Bhagyashri Wagh[*], [2]Prof. J. V. Shinde, [3]Prof. P. A. Kale

[1] M.E. Student, Department of Computer Engineering, Late G.N.Sapkal COE and Management, Nashik ,India

[2] Asst. Professor, Department of Computer Engineering, Late G.N.Sapkal COE and Management, Nashik, India

[3] Asst. Professor, Department of Computer Engineering, Late G.N.Sapkal COE and Management, Nashik, India

Abstract—

In today's world, Social Networking website like Twitter, Facebook , Tumbler, etc. plays a very significant role. Twitter is a micro-blogging platform which provides a tremendous amount of data which can be used for various application of sentiment Analysis like predictions, review, elections, marketing, etc Sentiment Analysis is a process of extracting information from large amount of data, and classifies them into different classes called sentiments. Python is simple yet powerful, high-level, interpreted and dynamic programming language, which is well known for its functionality of processing natural language data by using NLTK (Natural Language Toolkit). NLTK is a library of python, which provides a base for building programs and classification of data. NLTK also provide graphical demonstration for representing various results or trends and it also provide sample data to train and test various classifier respectively. Sentiment classification aims to automatically predict sentiment polarity of users publishing sentiment data. Although traditional classification algorithm can be used to train sentiment classifiers from manually labelled text data, the labelling work can be time-consuming and expensive. Meanwhile, users often use some different words when they express sentiment in different domains. If we directly apply a classifier trained in one domain to other domains, the performance will be very low due to the difference between these domains. In this work, we develop a general solution to sentiment classification when we do not have any labels in target domain but have some labelled data in a different domain, regarded as source domain.

Keywords— Sentiment Analysis, NLTK (Natural Language Toolkit), Python

I. INTRODUCTION

Sentiment analysis can be defined as a process that automates mining of attitudes, opinions, views and emotions from text ,speech, tweets, and database source through NLP it is also known as opinion mining, it determines whether a piece of writing is positive, negative or neutral. Today most of the people use social networking sites to express their opinion about something .companies have been receiving polls about the products they manufacture. the sentiment analysis is done using various machine learning technique. [1] analysis can be done at document, phrase and sentence level. In document level ,the entire document is taken then it is analyze whether the sentiment is positive, negative or neutral. in phrase level, analysis of phrases in a sentence is taken in account to check the polarity. In sentence level, each sentence is classified into number of classes. The goal of Sentiment Analysis is to harness this data in order to obtain important information regarding public opinion, that would help make smarter business decisions, political campaigns and better product consumption. The textual information retrieval techniques mainly focus on processing, searching or analyzing the factual data present. Sentiment classification has number of application which is helpful in business, marketing and increasing sell of the product people .in this we are doing the sentence level analysis, all the sentence are taken in to the .csv file then pre-processing apply on that sentences and using machine learning algorithm that data is classified. The paper organization is as follows. Section 2 shows the previous work done in this field .The steps of the algorithm and the flow diagram is described in section 3 and section 4 represents result of different datasets. Finally section 5shows the conclusion part[14]

II. RELATED WORK

A model to classify the tweets as objective, positive and negative proposed in[1]they created a twitter corpus by collecting tweets using Twitter API and automatically annotating those tweets using emoticons. Using that corpus, they developed a sentiment classifier based on the multinomial Naive Bayes method that uses feature like N-gram and POS-tags. The Naive Bayes bigram model and a Maximum Entropy model are implemented in [2]

To classify the tweets from this two model Naive Bayes classifiers worked much better than the maximum Entropy Model. A solution for sentiment analysis for twitter data by using distant supervision in which their training data consisted of tweets with emotions which served as noisy labels. They build models using Naive Bayes, MaxEnt and SVM.[3] Their feature space consisted of unigrams, bigrams and POS. They concluded that svm outperformed other models and that unigrams were more effective as features.

Three way model for classifying sentiment into positive, negative and neutral classes they experimented with models such as: unigram model, a feature based model and a tree kernel based model. For tree based model they represented tweets as a tree. The feature based model uses 100 feature and the unigram model uses over 10,000feature.They arrived on a conclusion that feature which combine prior polarity of words with their Parts-of-speech(pos) tags are most important and plays a major role in the classification task. The tree kernel based model outperformed the other two models.[4].

To utilize Twitter user-defined hash tags in tweets as a classification of sentiment[5]type using punctuation, single words,n-grams and patterns as different feature types, which are then combined into a single feature vector for sentiment classification. They made use of K-Nearest Neighbour strategy to assign sentiment labels by constructing a feature vector for each example in the training and test set. Twitter API to collect twitter data. then the data is labels as positive, negative and non-opinions. Tweets containing opinions were filtered. Unigram Naive Bayes model was implemented and the Naive Bayes simplifying independence assumption was employed. They also eliminated useless feature by using the Mutual Information and Chi square feature extraction method. Finally, the orientation of an tweet is predicted i.e. positive or negative. In[6] presented variations of Naive-Bayes classifiers for detecting polarity of English tweets. Two different variants of naive Bayes classifier were built namely baseline and Binary

III. SYSTEM ARCHITECTURE

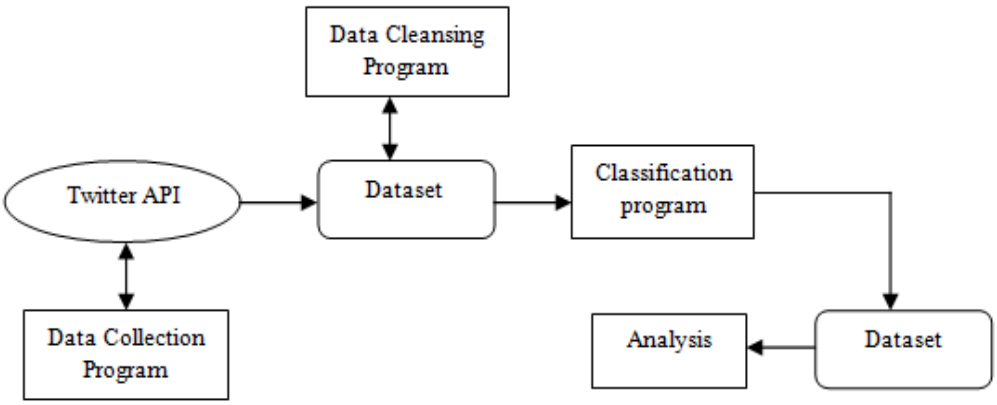

Fig. 1 System Architecture diagram

The proposed system is schematically shown in figure1. First we are going to stream tweets in our build classifier with the help of Tweepy library in python. Then we pre-process these tweets, so that they can be fit for mining and feature extraction After pre-processing we pass this data in our trained classifier, which then classify them into positive or negative class based on trained results. Since, Twitter is our source of data for analysis. We are going to stream tweets from twitter in our database. For this we are going to use Twitter Application

A. Twitter API /Data storage

Twitter allows users to collect tweets with the help of Twitter API. Twitter provides two kinds of APIs: Rest API and Streaming API. The difference Between these are: REST APIs support connections for short time

interval and only limited data can be collected at a time, whereas Streaming API provides tweets in real –time and connection for long time. We use Streaming API for our analysis. For collection large amount of tweets we need Long-lived connection and limit data rate. Once, we start getting our data from the .csv file our next step is to store data so that we can use it for sentiment analysis. We use .csv format for our collected data files because data consists of many fields.CSV separate each field with a comma, thus make it very to access the particular field which consists of text.CSV files also provide faster read/write time as compared to others.

B. Data collection

To use Twitter API we must first have a twitter account. it can be easily created by filling the sign up details in twitter.com website .After this you will be provided with a username and password which is use for login purpose .Once your account create you can now read and send tweets on any topic you want to explore. In this script we used the twitter dataset publicly made available by Stanford University. An analysis was done on this labelled datasets using various feature extraction technique.

C. Pre-processing data

Data obtained from twitter is not fit for extracting features. Mostly tweets consists of message along with usernames ,empty space, special character, stop words, emoticons, abbreviations, hash tags, time stamps, URL's ,etc. Thus to make this data fit for mining we pre-process this data by using various function of NLTK.IN pre-processing we first extract our main message from the tweet ,then we remove all empty spaces ,stop words(like is, a, the, he, them, etc),hash tags, repeating words, URL's etc. we then replace all emoticons and abbreviations with their corresponding meanings like,=D,=), lol , Rolf ,etc. Are replaced with happy or laugh. Once we are done with it, we are ready with processed tweet which is provided to classifier for required results. Sample tweet and processed tweet. Cleaning of twitter data is necessary, since tweets contain several syntactic features that may not be useful for analysis. The pre-processing is done in such a way that data represented only in terms of words that can easily classify the class.

We create a code in python in which we define a function which will be used to obtain processed tweet. this code is used to achieve the following functions. Other data which we collected for this thesis is training data. This data is used to train the classifier which we are going to build. To collect this data we use NLTK library of python. NLTK consist of corpora which is very large and consists of structured set of text files which are used to perform analysis In these corpora there are various types of text files like quotes, reviews, chat, history, etc From these corpora we will select fiels of movie reviews for our training purpose.

D. Classification data

To classify tweets in different class(positive and negative)we build a classifier which consists of several machine learning classifiers to build our classifier we used a library of python called ,scikit-learn. Scikit-learn is a very powerful and most useful library in python which provides many classification algorithm. Scikit-learn also includes tools for classification, clustering , regression and visualization to install scikit-learn we simply use online command in python is 'pip install scikit-learn''. In order to build our classifier, we use in build classifiers which come in scikit-learn library, which are

1. Naive-Bayes classifier

Naive-Bayes classifiers are probabilistic classifier which come under machine learning techniques. These classifier are based on applying Bayes' theorem with strong(naive)assumption of independence between each pair of features. let us assume, there is a dependent vector from $x1$ to xn and a class variable 'y'. Therefore, according to bayes: [8]

$$p(y|x1, \dots\dots, xn) = \frac{p(y)p(x1, \dots xn|y)}{p(x1, \dots . xn)}$$

Now according to assumption of independence

$$p(xi|y, x1, \dots\dots xi-1, xi+1, \dots\dots, xn) = p(xi|y),$$

For every i , this function becomes

$$p(y|x1, \dots\dots, xn) = p(y) \prod_{i=1}^{n} \frac{p(xi|y)}{p(x1, \dots . xn)}$$

In this $p(x1, \dots\dots, xn)$ on this input is constant, hence we can apply classification rule as:

$$p(y|x1,\ldots..,xn) \propto p(y)\prod_{i=1}^{n} p(xi|y),$$

$$\hat{y} = argmaxp(y)\prod_{i=1}^{n} p(xi|y),$$

And for estimating we can use MAP(Maximum A Posterior)estimation $p(y)$and $p(xi|y)$the $p(y)$ of class 'y' in training sample is relative frequency.

2. MultinominalNB classifier

MultinominalNB expands the use of NB algorithm. It implements NB for data distributed multinomially, and also uses one of its version for text classification (in which word counts are used to represents data, and also tf-idf works extremely well in regular practice). we parameterized the distribution data by vectors for every y, where 'n' gives the total feature(which means, the size of vocabulary for next classification) and probability $p(xi|y)$ of each 'i'that appears in the sample of class 'y' is θ_{yi} then use smoothed version of maximum likelihood for estimation of parametersθ_{y1} which is relative frequency of counting [8]

$$\theta_{yi} = (\theta_{yi},\ldots..\theta_{yn})$$

$$\theta_{yi} = \frac{Nyi + \alpha}{Nyi + \alpha n}$$

Where $N_{yi,}$ Represents number of times 'i' appeared in any sample of class y which belongs to training sample T and

$N_{yi,} = \sum_{i=1}^{T} N_{yi,}$ gives the total number of features in class 'y'. To prevent zero probabilities for further calculations, we add smoothing priors $\alpha \geq 0$ for features that are not present in any learning samples.

3. BernoulliNB classifier

BernoulliNB also implements NB algorithm for training and classification. It use NB for multivariate Bernoulli distribution of data i.e., there can be many features but each and every one is assumed to have a binary value or Boolean (true or false) variable. Hence, every class requires samples which have to be represented in binary value variables; also if any other kind of data is given then BernoulliNB can Binaries its input. which is different from MultinomialNB's rule, this rule directly punish any unavailability of feature i which behave like a feature of class y where as in multinomial it simply ignore if there is any non-occurring feature. [9][10]

The BernoulliNB decision rule is explained as:

$$p(xi|y) = p(xi|y)xi + \big(1 - p(i|y)\big)(1 - xi)$$

4. Logistic regression

Despite its name Logistic regression model but a linear model for classification. this model is also known by other names as Maximum-Entropy (MaxEnt)classification or log-linear. A logistic function is used in this model, where probability describe the outcome of single trial. The logistic regression can be implemented from Scikit-learn library of Python in which there is a class named Logistic Regression. This implementation fits a OvR (one-vs-rest) multiclass regression with an optional L1 or L2 regulation. L2penalized logistic regression helps in minimizing the following cost function. [11]

$$min_w, c\frac{1}{2}wTw + C\sum_{i=1}^{n} \log(\exp(-yi(x_i^T w + c)) + 1)$$

Similary,L1 regularized logistic regression can solve following problem of optimization

$$min_w, c||w|| + C\sum_{i=1}^{n} \log(\exp(-yi(x_i^T w + c)) + 1)$$

5. SGDC

SGDC is a simple yet powerful and efficient approach for learning of classifiers that comes under convex loss functions such as SVM and Logistic Regression. SGDC combines multiple binary classifiers in OvA (one-vs All) method. Therefore it supports multi-class classification. During testing phase, we also calculate confidence score for each and every classifier and thus choosing the class with the highest score. In multi-class, classification 'coef 's' a 2-D array of shape=[classes, features] and 'intercept' is a 1-D array of shape=[classes]only. The ith row of coef matrix contains the weight quantity for OvA classifier of the ith class. Also, classes are arranged in increasing order. [12]

6. Linear SVM

SVM are supervised machine learning methods used for classification, regression and detection models.SVM is more effective foe high dimensional space. SVCs are capable for multi-class classification. SVCand NuSVC is similar whereas, LinearSVC are based on linear kernels. All these SVCs take two input

array X of size [samples, feature] and array Y of size[samples]. NuSvc implements 'one-against-once' scheme for multi-class, hence it provides consistent interface with other classifier. Whereas, LinearSVC implements' one-vs-rest' scheme. [13]

IV. EXPERIMENTAL RESULT

A. Dataset Used

We used the twitter dataset publicly made available by Stanford university .Analysis was done on this labeled datasets using various feature extraction technique. We used the framework where the pre-processor is applied to the raw sentences which make it more appropriate to understand. Further, the different machine learning techniques trains the dataset with feature vectors and then the semantic analysis offers a large set of synonyms and similarity which provides the polarity of the content Once, we start getting our data from the csv file we can use it for sentiment analysis. We use .csv format for our collected data files because data consists of many fields.CSV separate each field with a comma, thus make it very to access the particular field which consists of text.CSV files also provide faster read/write time as compared to others. In the proposed system we have used the Stanford dataset i.e. 4million tweets categorized as positive and negative. since in the existing system they used the 45000 train data and 44832 Test data they used naive bayes as a baseline algorithm and we used another classification algorithm on different number of dataset. We studied that more the cleaner data, more accurate results can be obtained. following are some comparison result.

B. Performance parameter for evaluation

Table I Confusion Matrix

	Predicted Positives	**Predicted Negatives**
Actual Positive	**TP**	**FN**
Actual Negative	**FP**	**TN**

Accuracy= (TP+TN)/(TP+TN+FP+FN)
Precision=TP/(TP+FP)
Recall= TP/(TP+FN)
F1= (2* Precision*Recall)/(Precision+ Recall)

In which TP,FN,FP and TN refer respectively to the number of true positive instances, the number of false negative instances, the number of false negative instances, as defined in table I.

C. Results

Figures shows the accuracy of different algorithm for different number of tweets. the average accuracy of all algorithm is 50% in fig.2 shows the comparison of two algorithm i.e. naive-bayes and MultinominalNB the accuracy percentage of MultinominalNB is highest than naive-bayes for some number of tweets. Sometime the result is found same because of the tweets that we pre-processed are meaningless i.e. no features are extracted so it was difficult to calculate the accuracy. similarly in fig.3 and fig.4 we get accuracy of the LinearSVC, BernoulliNB classifier ,Logistic Regression and SGDC classifier respectively. The results of the proposed system is found more efficient than existing system.

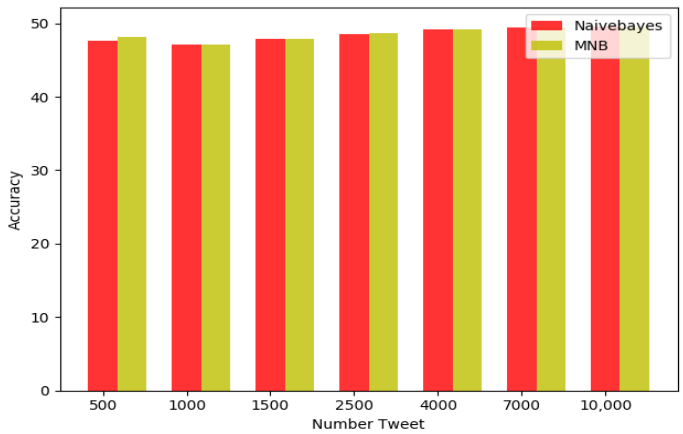

Fig.2 Graphical representation of naive-bayes and MNB

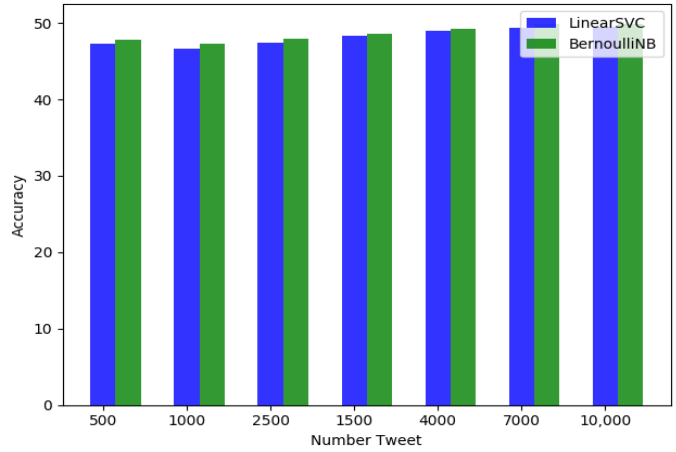

Fig.3 Graphical Representation of LinearSVC and BernoulliNB

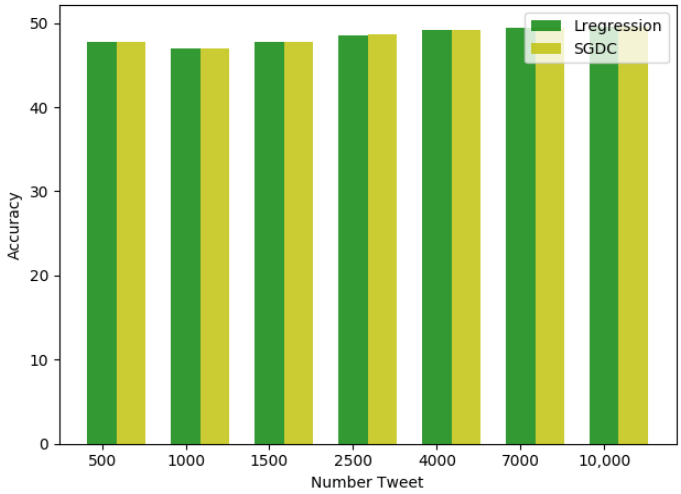

Fig.4 Graphical Representation of Lregression and SGDC

D. Comparative results

Table II Algorithm accuracy of existing system

Number of tweets	Naïve bayes	LinearSVC (SVM)	Logistic Regression
500	47.60	47.34	47.70
1000	47.11	46.67	46.94
1500	47.86	47.43	47.82
2500	48.45	48.29	48.56
4000	49.14	49.02	49.12
7000	49.45	49.39	49.43
10,000	49.61	49.54	49.61

Table III Algorithm accuracy of proposed system

Number of tweets	MNB Classifier	BernoulliNB	SGDC Classifier
500	48.17	47.77	47.80
1000	47.16	47.28	46.97
1500	47.86	48.00	47.71
2500	48.64	48.56	48.70
4000	49.19	49.19	49.23
7000	49.46	49.88	49.45
10,000	49.61	49.93	49.54

V. CONCLUSIONS

In this paper, The system can also computes the frequency of each term in tweet. Using machine learning supervised approach help to obtain the results. Twitter is large source of data, which make it more attractive for performing sentiment analysis. We perform analysis on dataset which is publicly available by Stanford University which contain total 4milion tweets, so that we analyze the results, understand the patterns and give a review on people opinion. We can conclude that more the cleaner data, more accurate results can be obtained. A web based application can be made for work in future. We can improve our system that can deal with sentence of multiple meanings we can also increase the classification categories so that we can get better results. We can start work on multi languages like Hindi, Spanish and Arabic to provide sentiment analysis to more local.

ACKNOWLEDGMENT

My sincere thanks go to KCTs Late G.N. Sapkal College of Engineering for providing a strong platform to develop my skill and capabilities. we would like to thanks all those who directly or indirectly help us in presenting the paper. I hereby take this opportunity to express our heartfelt gratitude towards the people whose help is very useful to complete our project. I would like to express our heartfelt thanks to my guide prof. J.V.Shinde and prof. P.A.Kale whose experienced guidance became very valuable for me.

REFERENCES

[1] Ke Tao, Claudia Hauff , Geert-Jan Houben, txing AG,TU Delft, ”Facilitating Twitter Data Analytics: platform, Language and Functionality".

[2] A. Pak and P. Paroubek,“Twitter as a Corpus for Sentiment Analysis and Opinion Mining,” In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010, pp.1320-1326 .

[3] R. Parikh and M. Movassate , “Sentiment Analysis of User-Generated Twitter Updates using Various Classification Techniques”,CS224N Final Report, 2009.

[4] Go. R.Bhayani , L.Huang,“Twitter Sentiment Classification Using Distant Supervision,” Stanford University, Technical Paper,2009.

[5] Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau ,“Sentiment Analysis of Twitter Data ,”In Proceedings of the ACL 2011Workshop on Languages in Social Media,2011 , pp. 30-38.

[6] Dmitry Davidov, Ari Rappoport.”Enhanced Sentiment Learning Using Twitter Hashtags and Smileys”. Coling 2010:Poster Volume pages 241{249,Beijing,Augest2010.

[7] Po-Wei Liang, Bi-Ru Dai ,”Opinion Mining on Social MediaData,” ", IEEE 14th International Conference on Mobile DataManagement,Milan,Italy,June3-6,2013,pp91-96,ISBN:978-1-494673-6068-5, http://doi.ieeecomputersociety.org/10.1109/MDM.2013.

[8] H.Zang, ”The optimality of Naive-Bayes”,Proc.FLAIRS,2004

[9] C.D. Manning, P.Raghavan and H. Schutze , ”Introduction to Information Retrieval”, Cambridge University Press,PP.234-265,2008.

[10] A.McCallum and K.Nigam,”A comparison of event models for Naive Bayes text classification”, Proc. AAAI/ICML-98 Workshop on Learning for Text Categorization,pp.41-48,1998.

[11] M. Schmidt, N.L Roux and F.Bach, “Minimizing Finite Sums with the stochastic Average Gradient”,2002

[12] Y.LeCun, L.Bottou, G.Orr and K.Muller, ”Efficient BackProp”, Proc. In Neural Networks: Tricks of the trade 1998.

[13] T.Wu, C.Lin and R.Weng, ”Probability estimates for Multi-class classification by pair wise coupling”,Proc.JMLR-5,pp.975-1005,2004

[14] “support Vector Machines”[online], http://scikitlearn.org/stable/modules/svm.html#svm-classification. Accessed jan 2016

[15] “An Introduction to Python”,v3.4.1, 2015[online], Available: https://docs.python.org.

[16] Pablo Gamallo, Marcos Garcia,“Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets,”", 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, Aug 23-24 2014, pp 171-175.

[17] Neethu M,S and Rajashree R, ",” Sentiment Analysis in Twitter using Machine Learning Techniques," 4th ICCCNT 2013,at Tiruchengode, India. IEEE – 31661.

[18] P. D. Turney, "Thumbs up or thumbs down: semantic orientation applied to unsupervised classification of reviews," in Proceedings of the 40th annual meeting on association for computational linguistics, pp. 417–424, Association for Computational Linguistics, 2002.

[19] Bifet and E. Frank, "Sentiment Knowledge Discovery in Twitter Streaming Data," In Proceedings of the 13th International Conference on Discovery Science, Berlin, Germany: Springer,2010, pp. 1-15.

[20] L. Barbosa, J. Feng ,"Robust Sentiment Detection on Twitter from Biased and Noisy Data, "COLING 2010: Poster Volume, pp. 36-44.

