JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

DESIGN AND ANALYSIS OF SENTIMENT ANALYSIS USING LEXICON BASED APPROACH AND NAIVE BAYES ALGORITHM: A BIG DATA RESEARCH PERSPECTIVES

Dr. SASIKUMAR GURUMOORTHY[1], Dr. PERIAKARUPPAN SUDHAKARAN[2], Dr.RAJKUMAR RAJASEKARAN[3]

[1] Professor, Computer Science and Systems Engineering, Sree Vidyanikethan Engineering College, Tirupati, Andhra Pradesh, India

[2] Professor, Department of Computer Science and Engineering, SRM TRP Engineering College in Tiruchirappalli, Tamilnadu, India

[3] Associate Professor, Vellore Institute of Technology, Vellore, Tamil Nadu. India

[1]sasichief@gmail.com, [2]sudhakaran79@gmail.com, [3]vitrajkumar@gmail.com

ABSTRACT:

The element of study which analyses people perspectives, attitudes, evaluations, sentiments, and ideas from produced language are referred to Sentiment evaluation and Opinion Mining. It is among probably the most energetic investigate factors in natural language processing also it's also usually analysed around information mining, Web mining, and then textual content mining. For the benefits of its to business and society as a whole, the research has sent out outside of laptop science over the management sciences in addition to social sciences. The regular usage and development of sentiment analysis could be associated with the development of social networking like evaluations, Twitter, micro-blogs, blogs, forum discussions, and social networking sites. Additionally, today, it's not so difficult for us to collect big amount of opinionated details from the electronic media because of the evaluation purposes. Opinions are typical to nearly every man activity and therefore are really important representatives of the behaviors of ours. Our perceptions and beliefs of reality, as well as the selections we create, are mostly trained on how others see and evaluate the world. And also for the very same reasons, every time we're in confusion and anytime it's a crucial choice we usually look for the views of others. In this paper, we're gon na examine 2 algorithms because of the twitter sentiment analysis and also determine that works better.

KEYWORDS: Tweets, Machine Learning, Lexicon based Approach, sentiment, polarity, sentiment score, big data

I. INTRODUCTION

The web is a huge repository of people's opinions from all over the world about a product or a service or an issue. With the advancement of technology and the ease in the availability and access to the social media sites, people have been relying on these sites to gather an idea about what the rest of the world thinks of that product before buying it. Be it a product or a service, one would always want to know about the reliability, performance, security etc., before purchasing it. While this can be done by reading all the reviews stated by the customers but let's face it, it is very tiresome and timetaking. To make this a simpler and an easy task research is being carried out for over 15 years now on a concept called sentiment analysis. Sentiment evaluation is an information style mining which dealings the tendency of individuals ideas over Natural Language processing (NLP), computational linguistics in addition to text examination, that're used to get and evaluate very subjective info coming through the complete - mostly similar sources and social media. The analysed information quantifies the general public 's reactions or sentiments for particular items, ideas or people and also expose the contextual polarity of the info. In simpler words, Sentiment Analysis is the procedure of deciding if a slice of writing is good, neutral or negative. It is likewise often known as opinion mining, deriving the viewpoint or maybe attitude of a speaker. There are a lot of approaches to carry out the analysis but regardless of which algorithm we use, there are certain problems that can't be eliminated. When the reviews are simple and straightforward, it would be easy to identify the emotion behind it. For example, let's take a few reviews into account to understand the impact of the problems we are going to face.

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

1.1 Positive Reviews: (Movie Reviews)

1. Brilliant direction, great movie.
2. I absolutely loved the movie.
3. Excellent acting.

1.2 Negative Reviews: (Movie Reviews)

1. Pathetic movie.
2. It was quite boring.
3. Awful acting.

Reading the above reviews, we can clearly tell that they are very straightforward and to the point. If we look at the adjectives in each of the sentences, we get the sentiment of the user. So, training the classifier by giving the polarity to each of these adjectives. Although this is the case when the reviews are easy to crack. A lot of times, we see that the customers write paragraphs to express their happiness and/or anger. In those cases, subjectivity is lost. Chances are they have gone off the topic. It is still a problem that can be solved. But the problems that haven't been solved with precision are

1. Sarcasm
2. Irony
3. Thwarting

Like we already know, sarcasm is one of the types that are not quite understood even when used in a face to face conversation. It still makes it easier to guess by the tone or gestures. Now, imagine how time taking and hard sensing sarcasm from a bunch of words would be. In this project, we have decided to carry out Lexicon based approach to detect sarcasm within the given reviews. There are a series of steps that are to be followed to conduct the experiment which are,

1. Gathering the dataset.
2. Pre-processing.
3. Training the classifier.
4. Evaluation.

Majority of the data acquired by the users is unorganized and not structured properly. This has interested a lot of NLP researchers to carry out their experiments on the same regardless of the difficulties to overcome. The way reviews are stated online are sometimes difficult for humans to understand given the usage of sarcasm and irony. Now getting an algorithm to do all the tasks would be even more difficult. We will discuss the challenges faced in this process in the further sections.

1.3 Dataset Explanation

The dataset we used in this research paper is tweets. Using the search_twitter() function, we gathered 1000 tweets for the classification. For the lexicon-based approach, we manually gathered the tweets after which the classification was done. However, for the Naive Bayes approach, the classification is done upon the entering the word in the shiny app. The latter is obviously an easier method since you can enter the word and you will get the reviews on that word. But that is not the case for the lexicon approach, you can only get one set of reviews at a time.

II. PROPOSED METHODOLOGY

We are going to use a combination of several techniques to carry out our experiments. The process includes data gathering, data cleaning and data classification. Data gathering involves dataset retrieval from all the online resources. Data Cleaning steps include phrase splitting, tokenization, stop term removing, lemmatization, spell

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

modification. The naive Bayes strategy tools a machine learning classification technique using an emoticon/hashtag handler, and also by sentiment scoring of viewpoint, words using the classify_emotion() as well as classify_polarity(). The primary goal of this effort is usually to determine the portion of wrong information which has been found by the algorithm therefore we are able to possess a comparative analysis of both algos. The first method would involve the gathering of the tweets from the Twitter API. All the algorithms are performed in R script. Once the tweets are gathered, create a function that preprocesses the tweets, Preprocessing is done using gsub. After the preprocessing is done, the next step is to calculate the sentiment of the tweets. For the lexicon-based approach, we create a function that checks for the no. of negative words and positive words in every one of the tweets and also scores the tweet. If the score is greater than 0, it is classified and positive and if it is less than 0, it is classified as negative and neutral otherwise. Depending on the score it can also be classified as extremely positive or extremely negative in case it's in excess of or perhaps the same as two or even less than comparable to -2 respectively. While for the naive Bayes algorithm, classify_polarity() function from the sentiment package has been used. This has been trained with a large twitter dataset with positive and negative sentiments. This function predicts the sentiment of the tweets based on the following formula,

P(*pos*/tweets) = [P(*tweets/pos) . P(pos)*] / P(*tweets*) and the following code

counts <=> count[[key]]

total <=> counts[['total']]

score <=> abs(log(counts/total))

scores[[key]] <=> scores[[keys]] + scores

The output is displayed as Positive/Negative/Neutral. We have used R and shiny to perform the analysis. The most important probability that would be needed for us is the positive score and the negative score based on which the entire statement would be categorized as positive/negative/neutral. If the likelihood of the occurrence of good phrases in a declaration is greater compared to the probability of the bad words, it will be termed as Positive Statement.

P(positive):

*P(positive|tweet) = P(tweet|positive) * P(positive)*

2.1 Probability of Positive P(Pos)

For the benefit of the example, we need to say there is 3 possible classes: neutral, negative, and positive which would mean the probability of occurrence of either of the possibilities is 0.333. Therefore P(Positive) = 0.333

2.2 Probability that the tweet is Positive P(tweets|pos)

To calculate this, we want an education conventional of twitters that remained previously categorized obsessed by the 3 classes. We tokenize the tweets first and then classify them later because it can't be told that we would find those specific tweets in that particular class. Hence to avoid the probability on the lower side, we prefer this method.

*P(tweets|pos) = P(TI|pos) * P(TU|pos) * .. * P(TN|pos)*

Somewhere TI to TU is altogether the words in the twitter.

P(TI|pos)

In order to establish the likelihood of a certain term dropping into the group we are evaluating, we will require the next out of the instruction set:

- The volume of times TI comes up in tweets that have been noticeable closely as great in the coaching set.
- The whole volume of terms of tweets that have been noticeable as fantastic in the coaching set.

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

III. SYSTEM ARCHITECTURE

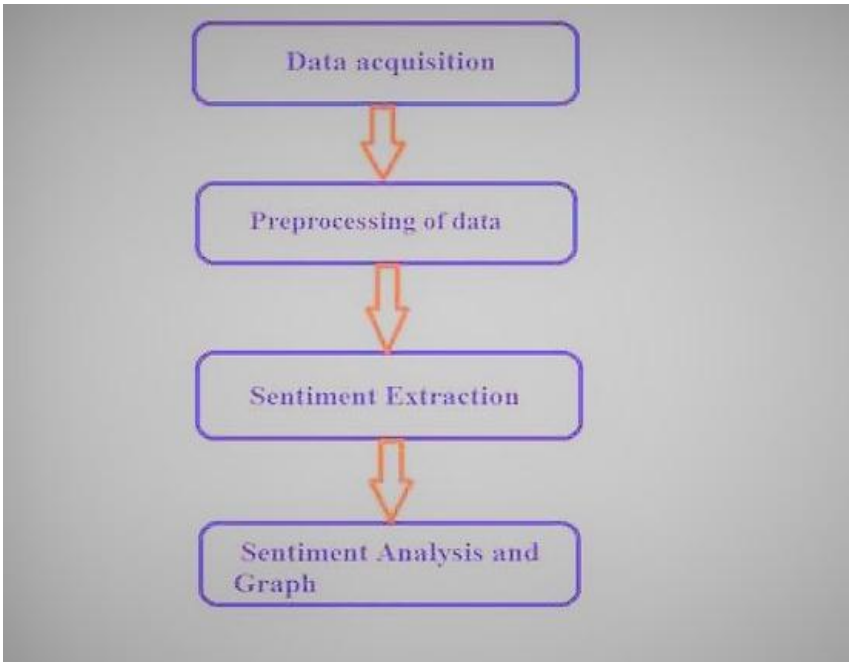

Fig 3.1

IV. COMPARISON OF THE ALGORITHMS

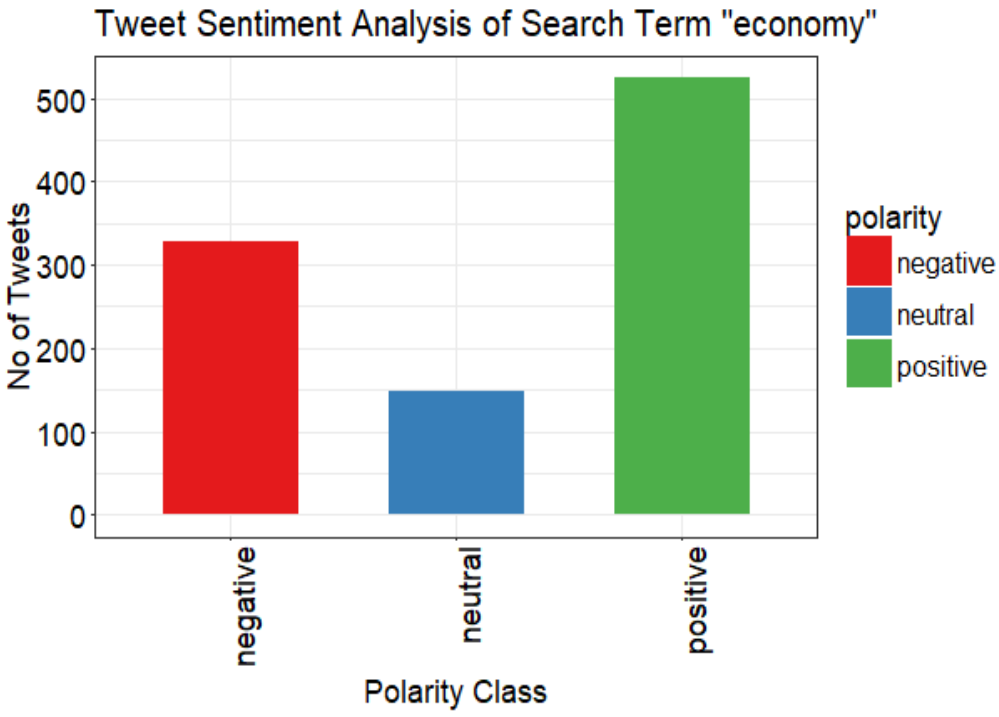

Fig 4.1

The above picture demonstrations the outcomes from the Naive Bayes' Approach for the tweets including #economy

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

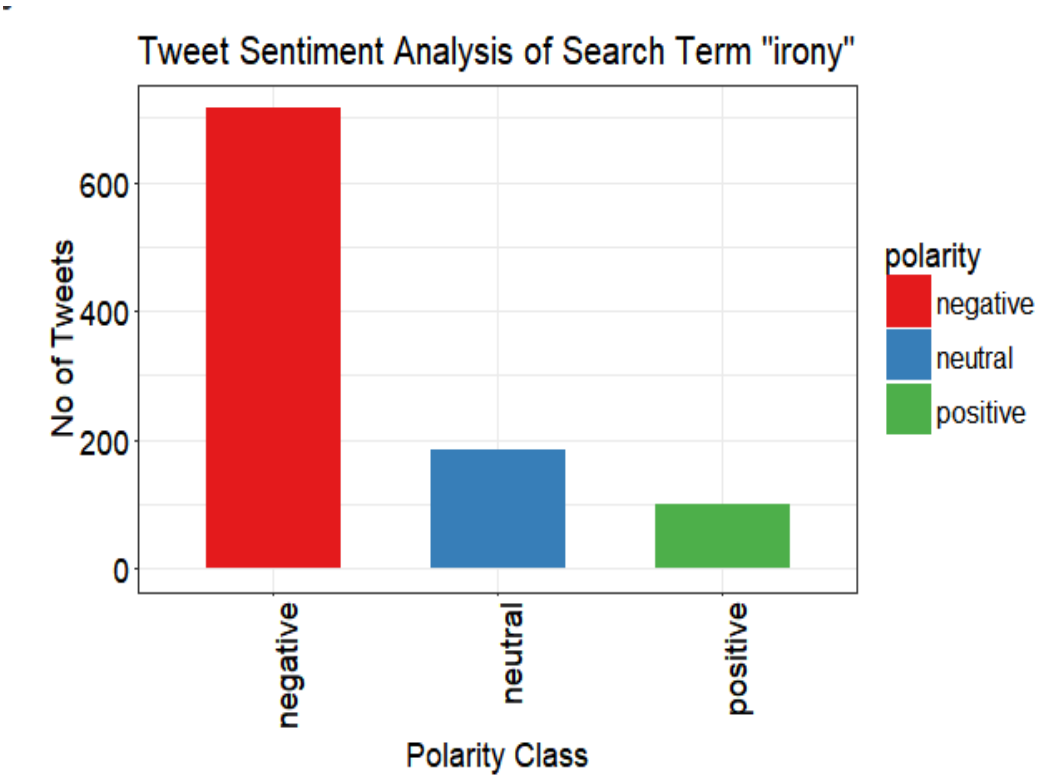

Fig 4.2

The above picture demonstrations the outcomes from the Naive Bayes' Approach for the tweets including #irony. It has been found that this algorithm has an accuracy rate of 75%. However it has been also seen that the results achieved from the lexicon based approach weren't as accurate as the Naive Bayes'.

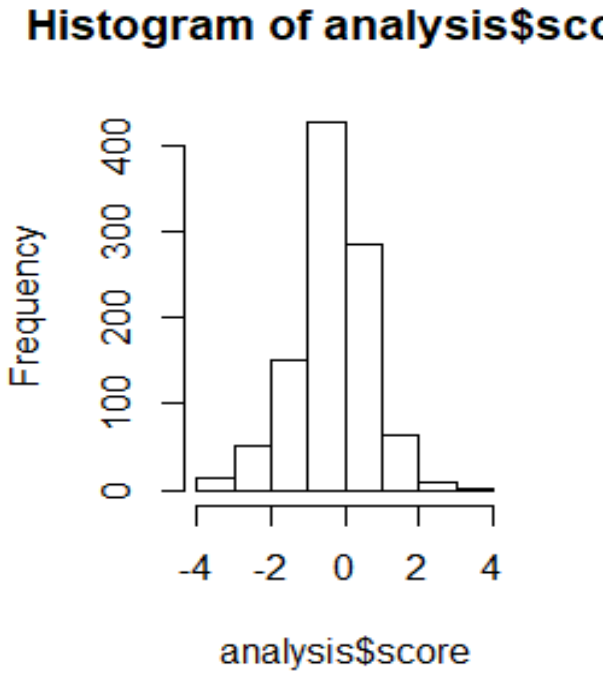

Fig 4.3

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

The above picture shows the results from the Lexicon based Approach for the tweets including #economy. As we can see maximum have been classified as neutral when almost 500 of them are classified as positive using the Naive Bayes' Approach.

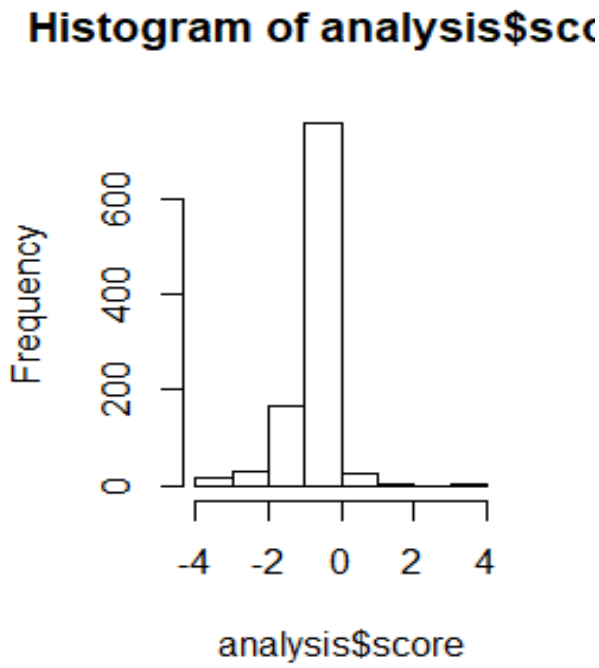

Fig 4.4

The above figure shows the results from Lexicon based Approach for the tweets including #irony. As we see that more than 80% of the data has been classified as neutral and using the Naive Bayes' we see that maximum tweets have been classified as negative. One of the problems with the sentiment analysis has been solved using the former approach.

V. EXPERIMENTAL RESULTS AND SETUP

We have used two classifier models to do the analysis. During the preprocessing phase, irrelevant information was removed from the dataset. The information includes the timestamp of when the records were submitted, the first row that contained retweets, comments, replies and emoticons. For the lexicon based method, we don't need a training and a testing dataset since the functions created work predict the sentiment of that particular tweets that have been streamed. We can use searchTwitter() to get the tweets we want and the function can be used to that. For comparison purposes, we have used same tweets on both the algorithms. However for the naive bayes, we have a huge dataset that is used for the training and the same training dataset is used for all the testing datasets, classify_polarity() was used to identify the polarity i.e., if the tweet is positive/negative/neutral. classify_emotion() is used to determine the emotion of that particular tweet. The training and validation accuracy of the Naive Bayes' was found to be 75% .

VI. RESULTS AND DISCUSSION

After training and testing the various classifiers which use the previously mentioned algorithms, it was found that the Naive Bayes' classifier was more accurate as compared to the lexicon based classifier.

VII. CONCLUSION AND FUTURE WORK

In conclusion, even though the Naive Bayes' classifier outperformed the lexicon-based classifier, it would be naive to say that the former is a better algorithm for classification as compared to the latter. The reason the authors feel this is due to the small dataset size. Hence to further understand and compare the two algorithms, it would be useful to obtain more data.

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

REFERENCES

1. Fernández-Gavilanes, M. Álvarez-López, T. Juncal-Martinez, J. Costa-Montenegro, E. & González-Castano, E. (2016), Unsupervised method for sentiment analysis in online texts, Expert Systems with Applications.
2. González-Ibánez, R. Muresan, S. & Wacholder, N. (2010)
3. Identifying sarcasm in Twitter: a closer look (2011)
4. Joshi, A. Bhattacharyya, P. & Carman, M. (2016), "Automatic sarcasm detection: a survey"
5. Pak, A. and Paroubek, P. 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining
6. Strapparava, C. and Valitutti, A. 2004. Wordnet-affect: an affective extension of wordnet
7. E. Lunando and A. Purwarianti, "Indonesian social media sentiment analysis with sarcasm detection" (2013)
8. B. Liu, "Sentiment analysis and opinion mining," Synthesis Lectures on Human Language Technologies (2012)
9. J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count (2002)
10. Utsumi, "Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from non irony (2000)
11. Gurumoorthy S, Muppalaneni NB, Sekhar C, Sandhya Kumari G. Epilepsy analysis using open source EDF tools for information science and data analytics. Int J Commun Syst. 2020;33:e4095. https://doi.org/10.1002/dac.4095
12. R. J. Kreuz and R. M. Roberts, "Two cues for verbal irony: Hyperbole and the ironic tone of voice (1995)
13. D. Tayal, S. Yadav, K. Gupta, B. Rajput, and K. Kumari, "Polarity detection of sarcastic political tweets (2012).