Detection of double negations in tweets using Machine Learning

Dildar Ahmed Saqib
saqib02@ads.uni-passau.de
University of Passau
Passau, Bayern

ABSTRACT

Good authors, comprehension writers and literature writers sometimes use double negatives in their writings. Therefore, to have an idea about the comprehension writing skills of a person or to know, which of the song writers are getting popular among young generation by using such type of double negatives and other rhetoric figures in their lyrics, we need to work on an algorithm that could detect such double negatives from a given text. In the current study we are going to design a pipeline through which we will be able to detect double negations in social media i.e., twitter. We will be implementing sentimental analysis to check the polarity of a sentence, text mining and various text processing techniques will be used to write its algorithm. We will compare how good is the sentimental analysis feature for such text detection. Moreover, after obtaining different features and applying different techniques contained in the algorithm, Multinomial Naïve Bayes classifier will be implemented to classify the tweets. When accuracy parameters were calculated, it was seen that the mentioned research is giving us better results than the ones already done in the field.

KEYWORDS

Double negations, twitter analysis, Naïve Bayes, sentiment analysis, text mining

ACM Reference Format:
Dildar Ahmed Saqib. 2018. Detection of double negations in tweets using Machine Learning. In *Proceedings of ACM Conference (Masterseminar Intelligente Systeme-19).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Double negatives are technically a grammar error in formal English language. Although, we come across them in song lyrics, literature, speeches and in our daily life. It can be compelling sometime to convey some information, sometimes it can be funny as well. Since, in our daily lives, we deal with most of the people in a formal manner, so we don't use double negative quiet often. There is not a lot of work done in the topic of implementation and understanding of sentimental analysis approach for the detection of double negative. Moreover, while doing the same for some online texts that has been collected live, is actually a problem.

Masterseminar Intelligente Systeme-19, January 2020, Passau, BY, Germany

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

Therefore, to guess the polarity of a sentence and to judge the speech/writing capabilities of a group of persons we need to implement an efficient technique that could do this job.

Secondary goal of the paper is to know how the sentimental analysis are done and to learn the use of various libraries of python e.g., TextBlob, CountVectorizer, that will be used in the algorithm and then apply text processing and machine learning techniques throughout the algorithm. Ali and his fellows presented a research where they proved that TextBlob is a better analyzer as compare to other sentimental analyzer techniques such as SentiWordNet and W-WSD [9]. Walid [11] also worked and compared different analyzer for "theShukran" social network for sentimental analysis, on comparing one can say that TextBlob is better than any other technique used for sentimental analysis.

2 RELATED WORK AND LITERATURE REVIEW

Amna and Tanku presented their work on the identification of negations and their semantic understanding in texts. They also tried to improve the existing sentimental text analysis approaches [2]. They presented a framework which had five parts. In the first part name as "Preprocessor", the data was sent as input, here the input data was refined in to the form of text and sentences to apply different text processing techniques afterwards. In the next step, the textual data was sent to the syntactic parser where the texts were tagged by parts of speech (POS) using Penn Tree Bank[13]. In the third and main step of the algorithm, they sent the data to the sentiment analyzer. Here the polarity of each of the sentence is calculated using a polarity calculator and a score is saved.

Choi and her colleague [5] proposed a two-step process for compositional semantics. In the first stage, they calculated the strength of a each in a given sentence and in the next step the polarity modification features was taken into account. For example in the sentences:

- They could not vanish all the risks in the system.

Here "not" which is a negater, negates the polarity of vanish. These rules and methodologies differ much when we compare them with the ones in [20], [14] and [18]. Similarly, Karl and her colleagues worked on vector based approaches to know the semantics of logical operations. They considered the problem as linear and gave more importance to basic design model. They updated the existing MV-RNN model to check the effects of different negations in the texts [10]. They worked on vector based representation of semantics, they also focused on negation on higher dimensional distributional models.

Martine, Erik and Lilja designed a toolkit for the detection of negations in context. They achieved it by using Maximum-margin classification [7]. They obtained the Conal corpus for training and testing purpose. Then, focused on the scopes and cues from the obtained corpus. We can understand their algorithm by the following example:

- And despite all his efforts, he was not the first one.

Here "not" is the cue, sometimes, the prefixes like use of "un" in the beginning of a word can also be considered as cue. Lets take the following example:

- He was the one of the 15 unfortunate ones that were left on ship.

Here the prefix "un" in word unfortunate is the cue. Sometimes such prefixes are present in the word but those words are not considered as negation words, such as in the following sentence:

- He was understanding my speech keenly.

In the above sentence, no matter there is an "un" as prefix but it is not a negative sentence. There are few other prefixes which are not taken as negative words, such as "immigration" is a general word but "immortal" is negative word. So, it is relatively difficult to deal with such words when working on negation detection especially double negation detection on English texts. For each sentence labelling was done for each cue and scope present in a sentence. [7] explained that cues are the negative words in a sentence, and scopes are the set of words whom present cue is effecting.

Eduardo Blanco and Dan Moldovon presented a paper in which they explained how important the focus and scopes are in a negative sentence. [4] Various forms in which a negation could occur in a sentence were defined. Some comparison between clausal, analytic, verbal and ordinary negations were made in the context of negation statements. In the following lines, some of their comparisons can be read:

- Verbal negation
  We can not buy anything at all.
- Analytic negation
  He did not do his task properly.
- Clausal negation
  They don't have much time left.
- Ordinary negation
  He didn't go to the market, he couldn't make it.

They used pseudo-relation for for addition of semantic relation in a sentence, Penn Tree bank was used for this purpose.

In some other research done by Laura, Raffealla and Raquel, relation between negated adjectives and antonyms was studied with the method of Distributional semantics [1].

3 APPROACH

The techniques of sentimental analysis will be used after getting the tweets from social media API tweepy. To design new apps for twitter and to develop new products, twitter launched their tweepy api for developers. Afterwards, the tweets will be sent to algorithm and finally to the machine learning part where Multinomial Naïve Bayes classifier is being used to predict whether the tweet contains double negation.

The complete flow of algorithm is being shown in Figure 1.

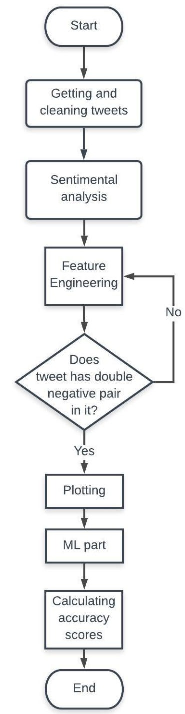

Figure 1: Flow of pipeline

3.1 Importing tweet data from tweepy API

User authentication of tweepy API is done using consumer key and consumer key secret. In the next step, search terms are defined to be used in search function in order to fetch tweets containing words like no, not, cant, none etc. [17] told us that increasing the number of features for the tweets increases the efficiency of the classifier.

3.1.1 Preprocessing. These search terms are then passed to get 100 tweets containing per single search term. The search function is called four times with different search terms, that means 400 tweets, after removing retweets 150+ tweets are left. Multiple calls are made to get a bigger data set which will contain roughly 400 tweets including the retweets.

3.1.2 URL/username removal. The fetched tweets are then initially preprocessed to remove tweet *urls* linked with the obtained tweets. Many tweets will be retweets of some other tweets, since we need only distinct tweets to apply the algorithm, therefore these retweets must be ignored and removed. Retweets, duplicates, special characters, unwanted data etc should also be removed since before sending the tweet sentences to the algorithm, the tweet sentences must be cleaned. We are not concerned with user name linked with a tweets, it also has to be removed.

3.1.3 Text lowering. Some other text cleaning actions are also involved like trimming, lowering the text etc to get a clear and text-only tweets. There were also some unnecessary white spaces between the tweet words, those spaces also need to be removed.

3.2 Main pipeline

Different stages involved in the algorithm are discussed in the following subsections:

3.2.1 Analyzing Tweet Sentiment . Tweet text is analyzed using TextBlob. The received tweets will be analyzed in first phase and then classified as positive or negative depending on the sentimental library decision. The tweets are classified as ‘negative’ or ‘positive’ based on the value of polarity given by TextBlob analysis i.e. > 0 is positive and < 0 is negative. Sentimental analysis only gives a value between -1 and 1 telling the chances of it being negative or positive. It’s also a machine learning model having a huge number of libraries and dictionary. It checks the sentences for positive words like "good", "should", "yes" etc and assigns weights and values to them depending on weight of each word in a sentence, overall polarity of the sentence is calculated.

Pablo and Marcos worked on a method through which they could also distinguish neutral tweets using polarity lemmas[8]. In the next stage, Tweet data is stored in *pandas* Dataframe. The tweets and their respective sentimental polarity verdicts were stored in a Dataframe. Then, the tweet data is stored in Dataframe. Tweet text, which is a string, is stored in a list named tweets_words Total number of words in string is calculated and then stored in Dataframe as ‘total_words’. We have calculated the number of words in a tweets to verify that the tweets containing negations have more number of words in them as compare to normal tweets.

3.2.2 Feature engineering . A dictionary containing possible negative words is declared. Also the one, with words having double negatives pairs in it, is defined. We declare a dictionary of negative word pairs which make a sentence double negative. This step is often referred as feature engineering.

These negative words and pairs play a very important role in decision for polarity and in sentimental analysis for a given tweet[3]. Michael Wiegand and his colleagues also worked on knowing the part, the negative words play in sentimental analysis[19].

3.2.3 Comparing Tweet with dictionary. Then we count the number of negative words in tweets by comparing the tweet’s word with negative words defined as in the dictionary. We store the count value that how many times, negative words were present in a tweet as ‘negative_words’ in the Dataframe. In the Table 1, an example of one such data frame is shown:

We also then compare the tweets one by one with double negative pairs dictionary to check if the tweet text has a double negative present in it. The final shortlisted tweets would now only include those which have double negative phrases in them. Then we display these tweets where double negative were detected.

A histogram was developed to see the relation between number of tweets containing double negatives and their polarity given by sentimental analysis part. We see that the most of the times, the sentiment analysis gave us "positive" verdict. Figure 2,3 and 4 are showing us that mostly sentences containing double negatives in them were labelled as "Positive" by the sentimental analysis method. The results of plots obtained from tweets are in various real times. We can see from the figures that most of the double negative tweets have positive sentiment.

Sr. no	tweets	sentiment	total_words	negative_words	is_double_negative
0	he has not paid for that one that us why...	positive	23	1	False
1	really not looking great for jo swinson you re...	positive	21	1	False
2	actually he seems like someone who wouldn t be...	positive	38	2	False
3	no it does not its for puddings	negative	23	1	False
4	ylikes maybe not then i just like warmth	negative	8	1	False
5	he has not paid for that one that us why...	positive	23	1	False
6	4 receivers leave for the nfl mecole votes for...	negative	49	1	False
7	can t wait to have my own car amp own place de...	negative	47	1	False
8	i am also angry that you clinton people can t g...	negative	45	1	False
9	so you are saying lacazette has now no place in...	positive	19	1	False
10	you are out of order you cant handle the truth...	positive	38	1	False
11	don t call me boo i m not your bae	negative	10	2	False
12	carom billards bad position solved by jaime...	negative	44	1	False
13	forever and not just in 2020	negative	6	1	False
14	this is illogical i am talking about black men...	negative	38	1	False
15	i bet your stomach is not happy with you right...	negative	11	1	False
16	the translation is not correct but i will take it...	negative	12	1	False
17	really not ready for gyu357 first post after d...	positive	9	1	False
18	not every minot relationship is abusive...	positive	24	3	False
19	totally with you phil grew up watching the bdo...	negative	8	1	False
20	cuz i was sick of the last trillion dollar sev...	negative	29	1	False
21	really not surprising considering how the staf...	negative	23	1	False
22	actually i shouldn t laugh i am not the ice ma...	negative	52	2	False
23	we would not have loved sana if she listened t...	positive	33	2	False
24	i think i vary it quite a bit current wip is s...	positive	46	1	False
25	i see what you are doing here assuming these d...	positive	44	1	False
26	not ring sound like a cc song why do i always ...	positive	12	1	False
27	you niggas spend too much time on captions not...	positive	13	1	False
28	i agree with you in addition to this maybe tho...	negative	25	1	False
29	youre not very bright trump acted against inte...	negative	14	1	False
30	manic is on the way but it will not coming to ...	positive	23	1	False
31	no i didnt and they not even letting ppl call ...	negative	14	2	False
32	when you re active but not hunterrowland	negative	7	1	False
33	not how it works its a little thing called due...	negative	11	1	False
34	what was the point of killing the general why ...	negative	28	1	False
35	not a bad shout that	positive	5	1	False
36	it does because you know what i m not even abo...	positive	17	1	False
37	i really wish certain tv shows and so on did n...	positive	23	2	False

Table 1: Complete Dataframe with features

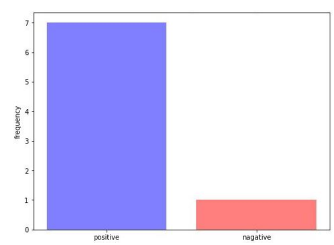

Figure 2: Tweet negations & Sentiment Analysis verdict

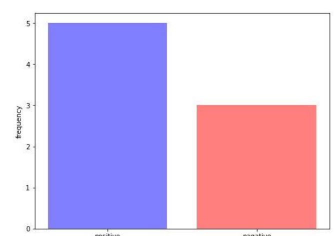

Figure 3: Tweet negations & Sentiment Analysis verdict

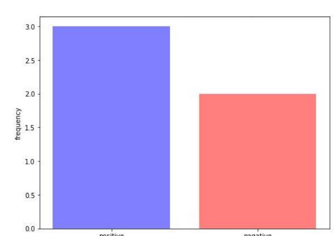

Figure 4: Tweet negations & Sentiment Analysis verdict

4 ML TECHNIQUE

In the following section, steps leading to the machine learning algorithm will be discussed.

4.1 Data preparation

In the first step, we reduced the amount of non-double negative tweets to increase the density of double negative tweets in the data.

We are removing some non-double negative tweets to make data better. The reason behind this i0,s with equal amounts of both types of data, the model will be able to train better in recognizing them. The data had too much non-double negative tweets which would have created confusion and disruptions within the model training. In other words, we increased the percentage of double negative tweets by removing some non-double negative ones.

Doing this has no effect on predictions as we are making an agent which takes data learns its features and predict future data based on what it has learnt, it learns better if there are comparable number of samples of each label. We achieved this by implementing following little steps:

```
totalRemoved = 0
for index, row in df2.iterrows():
    if (row['label'] == 0 and  totalRemoved(len(df1)-
4*len(dnegative_tweets))):
        totalRemoved += 1
```

It's a loop that goes through the rows of data frame. If label is 0 meaning it is non double negative, so it is removed. We keep removing till we have twice as many non-double negatives as doubles negatives until we get comparable number of double negative tweets and simple tweets. Afterwards, the job of storing the data in Dataframe, as tweets labelling is done. This is because storing data in this form is necessary for Naïve Bayes Classifier as it is designed to receive data in this format and to use it for training and testing in next stages. Training Data and testing data are both chosen by the model itself randomly. Then. we create a vector of words count i.e. rank the words based on their frequency of use in their respective label. We afterwards. split the data into training and testing sets. The algorithm takes 70-30 split for training and testing sets respectively.

CountVectorizer creates an array of words and their frequency. It's also called bag of words e.g., common words in all the tweets are stored in this array with their frequency. Andrew Ng and his colleagues also worked on similar approach with the help of unsupervised model for the incorporation of sentimental information[12]. Bag of words is an approach used in sentimental analysis where each word in the document is assigned a weight. Das and Chen[6] tried to extract negations from the Stock Market boards. Abinash, Ankit and Santanu [16] also used CountVectorizer technique in their research for classification of reviews using Sentimental analysis. They used it to make a vector, depending on the features present in a movie review. They then sent the data to the machine learning part. We now display vector of word count to show which words are used most frequently in both types of sentences. An example of such a vector obtained from the tweets is shown in Table 2.

The algorithm compares these words with the rest of the sentence to judge their type.

words	repetitions
don	6
stop	5
ain	5
s	4
haven	4
like	4
loving	3
tell	3
sora	3
arent	3
world	3
right	3
tht	2
said	2
things	2
personal	2
come	2
ppl	2
city	2
kairi	2

Table 2: Words obtained from CountVectorizer and TF/IDF

4.1.1 *TF/IDF formulation.*

Term Frequency Inverse Document Frequency is the basic block for various text mining and search algorithms. It simply calculates how many times, a certain word has appeared in a text or a given corpus. So here, we already have a data frame containing labels mentioning double negatives and non-double negatives and through this TF/IDF, we will get some features/repetitive word present in the tweets and based on these features, the Naïve Bayes algorithm will predict whether the coming tweet should actually be classified as double negative tweet not.

5 EVALUATION

In the following lines, you can see the result of tweets obtained at an instance.

(i) i ain t no hoe
(ii) that city glimmer ain t got nothing on this spotlight stealer keep the attention on you in the commander s chair of the lexusgx460 wednesdaywisdom
(iii) workin workin workin ain t ya u ain t got no time to lay up
(iv) when someone s trying to pick a fight with you so that they can take their unreleased anger out on you ain t got nothing to do with me next
(v) stop feeling bad for people when you tell it to them how it is and they get upset if you want a fairytale then don t come to me this world ain t no fucking fairytale
(vi) yo this ain t no laughing matter
(vii) i haven t had no junk food in two days
(viii) i been losing weight to fast i haven t had no appetite
(ix) i personally love mine it gave my some discomfort during men s trap cycles when i first got it but other than that it s

great i haven t had an actual bloody period since 2016 amp ain t no playing jenga in this lady

(x) the struggle is real i haven t had no sex since last year shit i barely was getting it then.

5.1 Sentiment's predictions

In figure 5, we can visualize how the sentimental decisions vary depending on the number of negatives in a sentence. In most of the cases, a tweets has a positive sentiment if it has even number of negation words in its sentence and gave us negative sentiment result when the tweets had odd number of negative words in them.

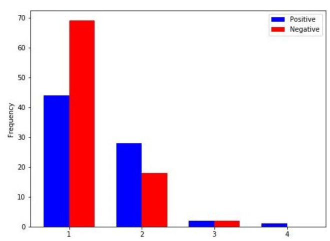

Figure 5: Type of tweet depending on number of negative words

5.2 Performing the predictions

Here a naïve_bayes object is defined using the MultinomialNB() function.

5.2.1 Multinomial Naïve Bayes. This is the formula for simple Bayes theorem, we are using Multinomial Naïve Bayes theorem in our algorithm.

$$P(C \mid F) = \frac{P(F \mid C)\, P(C)}{P(F)}$$

In the above mentioned formula C is the class to which a particular value belongs, and F is the feature used to discriminate the objects in different classes.

P(C/F) is the probability that a given sentence is double negative, given the negative words present in it. This probability is calculated by checking the presence of these words in prior sentences (our training data) and the outcome (whether it was a double negative or not). Based on this probability, we predict whether the sentence we have, is double negative or not. Alexander and Francesc used the same Multinomial Bayes theorem approach and worked on corpora obtained from twitter[15]. With their algorithm and approach, they achieved a maximum precision score of 62%.

We then fit the training sets into the model and in next step, we predict the results by passing the test set into it.

5.3 Displaying the result of model's prediction

We then use confusion matrix in the form of heat map. Confusion matrix is used to tell us how accurate our classifier is. It has four entries. True Positives tell us how many tweets were detected as Double negatives , provided they were actually double negative, similarly True Negative entry shows us how many tweets were classified as non-double negative tweets, given that were were

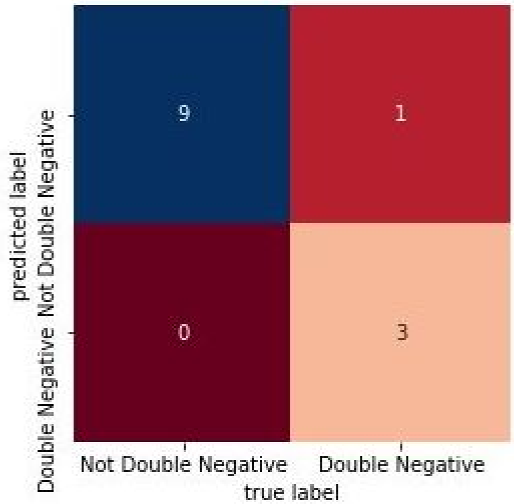

Figure 6: Actually obtained confusion matrix

actually normal(non-double negative) tweets. False Positive and False Negatives are the entries that tells how bad our classifier was.

Since most of the people don't use double negatives therefore we could only get a small number of double negatives in our input corpus/data set. We tried to increase the original data set size but there wasn't any considerable change in the obtained double negatives from those tweets. The reason behind this is, most of the users consider it difficult to write a sentence having double negative in it, because that makes the sentence more complex and require comparatively more English language command.

If there is an entry in one of these i.e., False negatives or false positives, it means, classifier was not able to predict a tweet correctly. In Figure 6, you can see a generic version of confusion matrix. If we also have False positive in our end result, this means that accuracy is going to be low since the algorithm judged a tweet as double negative tweet, although, it was not having double negative in it. Figure 7 shows one result of confusion matrix obtained at an instance.

A comparison display of actual vs predicted results was made. In the end we observed which of the tweets were counted as true positives and which were as true negatives to read.

In the last stage the data was prepared for machine learning part and then implementation of Naïve Bayes approach was made. We can also obtain a final confusion matrix after the algorithm completes its whole process for 100+ tweets. The tweets data that was used in training and testing purpose can also be viewed.

Naïve Bayes environment was created to apply NB algorithms for finding the evaluation parameters. Then, fitting of the training sets into the model was done. Prediction of the results was done by passing the test set into it. Since we don't have one particular corpus, the tweets keeps on changing at every attempt. Every new tweets have new sentence structure in it, therefore, the same algorithm gives us different accuracy score. Since all the tweets have now been fetched to the algorithm and double negations detection has been made, now it is the time to calculate how much accurate and precise our system is. For that we find out the accuracy, precision and recall scores. Most of the times, the accuracy scores were above 85% percent, i.e., our algorithm was efficiently distinguishing double negative tweets from the tweets corpora.

Precision tells us how exact the classifier was in its prediction and classification. Below is the formula for precision:

$$\text{Precision} = \frac{TruePositive}{TruePositive + FalsePositive}$$

Similarly, recall tells us how complete out classifier is.

In the final steps, good precision and recall scores were obtained based on the double negatives tweets which when manually read had the sentiment as TextBlob gave us in the first place. Most recent accuracy, precision and recall scores observed were 0.9, 1.0 and 0.5 respectively.

6 RESULTS AND DISCUSSION

Sentimental analysis technique i.e., TextBlob gave us good results, the tweets with even number of negatives were having positive sentiments and on the other hand, ones with odd number of negative words were considered as negative. We saw that when number of negative words in the tweet were even i.e., 2,4 the sentimental analysis verdict was positive for most of the tweets, on the other hand, it was counted as negative tweet when number of negative words in tweets are odd.

There are limitation in API as well, we could get thousands of tweets in start so that we have hundreds at the end but requesting huge data from API is not possible. A good accuracy, recall and score means that the defined algorithm was efficient to discriminate the double negative from the corpora and were better as compare to ones presented by [15]. There are some words which does come with prefix *un, ir,im* in them but these words are not every time considered negatives words such as *university, iron, impression.* Such words, when found in such algorithm can effect the training model.

7 CONCLUSION AND FUTURE WORK

Our presented algorithm worked well, accuracy scores were also good. We tried running several times, but a noticeable drop in these scores was not observed. Of course, these evaluation scores keeps on changing because after every run, program finds out new tweets and creates new data for machine learning. Sentimental analysis, text blob, various text mining techniques were learnt and implemented. Multinomial Naïve Bayes classifier performed well because we had prepared the data before sending it to classifier. Preparation of data was necessary because otherwise, the classifier won't had enough data from both set of tweets for its training. After applying this algorithm, we can detect which of the users have the ability to correctly generate double negatives (i.e., they are having good English composition skills) by tracing back those tweets from the obtained tweets data set.

Future work could contain updating the existing sentiment approaches and making it better for handling more complex sentences. Many new informal grammatical words such as *gonna, wanna* are invented every now and then, these words need to be added in the sentimental analysis approach to get the better results in the detection of double negative in tweets and other text corpora.

REFERENCES

[1] Laura Aina, Raffaella Bernardi, and Raquel Fernández. A distributional study of negated adjectives and antonyms. CEUR Workshop Proceedings, 2018.

[2] Amna Asmi and Tanko Ishaya. Negation identification and calculation in sentiment analysis. In *The second international conference on advances in information mining and management*, pages 1–7, 2012.

[3] Hamid Bagheri and Md Johirul Islam. Sentiment analysis of twitter data. *arXiv preprint arXiv:1711.10377*, 2017.

[4] Eduardo Blanco and Dan Moldovan. Some issues on detecting negation from text. 01 2011.

[5] Yejin Choi and Claire Cardie. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the conference on empirical methods in natural language processing*, pages 793–801. Association for Computational Linguistics, 2008.

[6] Sanjiv R Das and Mike Y Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management science*, 53(9):1375–1388, 2007.

[7] Martine Enger, Erik Velldal, and Lilja Øvrelid. An open-source tool for negation detection: a maximum-margin approach. In *Proceedings of the Workshop Computational Semantics Beyond Events and Roles*, pages 64–69, Valencia, Spain, April 2017. Association for Computational Linguistics.

[8] Pablo Gamallo and Marcos Garcia. Citius: A naive-bayes strategy for sentiment analysis on english tweets. In *Proceedings of the 8th international Workshop on Semantic Evaluation (SemEval 2014)*, pages 171–175, 2014.

[9] Ali Hasan, Sana Moin, Ahmad Karim, and Shahaboddin Shamshirband. Machine learning-based sentiment analysis for twitter accounts. *Mathematical and Computational Applications*, 23(1):11, 2018.

[10] Karl Moritz Hermann, Edward Grefenstette, and Phil Blunsom. "not not bad" is not "bad": A distributional account of negation, 2013.

[11] Walid Iguider and Diego Reforgiato Recupero. Language independent sentiment analysis of the shukran social network using apache spark. In *Semantic Web Evaluation Challenge*, pages 129–132. Springer, 2017.

[12] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[13] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. 1993.

[14] Nicolas Nicolov, Kalina Bontcheva, Galia Angelova, and Ruslan Mitkov, editors. *Recent Advances in Natural Language Processing III, Selected Papers from RANLP 2003, Borovets, Bulgaria*, volume 260 of *Current Issues in Linguistic Theory (CILT)*. John Benjamins, Amsterdam/Philadelphia, 2004.

[15] Alexandre Trilla and Francesc Alıas. Sentiment analysis of twitter messages based on multinomial naive bayes. *Comput. Surv*, 34:1–47.

[16] Abinash Tripathy, Ankit Agrawal, and Santanu Kumar Rath. Classification of sentimental reviews using machine learning techniques. *Procedia Computer Science*, 57:821–829, 2015.

[17] M Vadivukarassi, N Puviarasan, and P Aruna. Sentimental analysis of tweets using naive bayes algorithm. *World Applied Sciences Journal*, 35(1):54–59, 2017.

[18] Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. A survey on the role of negation in sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, NeSp-NLP '10, page 60–68, USA, 2010. Association for Computational Linguistics.

[19] Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. A survey on the role of negation in sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*, pages 60–68, 2010.

[20] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics*, 35(3):399–433, 2009.