International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-5, March 2020

Sarcasm Revealing using Rule Based Algorithm

S. Maheswari, K. Arthi

Abstract: ***Sentiment analysis is the process of finding out whether one's opinion is positive, negative, or neutral. Now-a-days the people are telling their opinion about the fields like marketing product, political and social phenomena are mostly through the online. Their opinions are positive, negative or neutral. The machine to identify the opinion is very difficult. There are so many issues in this field. The one of the issue is sarcasm detection. Sometimes the people give their opinion sarcastically. Sarcastic means, an opinion of an object is to say positive instead of negative. The machine will take this opinion as positive. So the final polarity of the product will be wrong due to this kind of identification. The purpose of this paper is to find these types of sentences and correct the polarity value.***

Keywords: Keyword: - Sarcasm, Lexical, Sentiment Analysis, Product Reviews

I. INTRODUCTION

Sentiment analysis is the field that study people's by the way in which something is regarded, a settled way of thinking, and a strong feeling deriving from one's circumstances of the text. This is the most prevalent presence in data mining, web mining and text mining at present. Data Mining cite to extracting the necessary information from large quantities of data. Web Meaning is one of the sub-domains of Data Mining, and it takes the necessary knowledge from WWW. This web mining breaks down three more domains, which are as follows:

i) Web Usage Mining ii) Web Content Mining iii)Web Structure Mining

The information in the analysis of sentiment is of importance in the form of text data. So the information is extracted from the web for text mining. Text mining is the process of deriving high-quality information from text. A high-quality text is one that combines relevance, novelty and interesting. Typically text mining tasks include the following: Text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization and entity relation modeling etc. Text analysis includes information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging, Predictive Analytics. Sentiment analysis study has many challenges. One of those is to detect sarcastic sentences. If the sentence is sarcastic, the polarity of the sentence in the sentiment analysis is incorrect; this means that the opposite polarity was intended. Sarcasm is a form of talk is what speakers say their message in implicit way. Even the humans cannot properly understand whether a sentence is a simple sentence or not. So it is important to find out whether the review of customers' products is sarcastic or not in development of Sentiment Analysis.

II. RELATED WORK

Today the researchers are very interested in finding the sarcastic in the text. All the data required for this research can be taken from Amazon Products Review, Tweets from Twitter, website comments and Google books. Sarcastic sentence is identified on the basis of some key features such as pragmatics, lexical hyperbole. The types of sarcasm occur can be detected on the basis of analysis of features. Generally features can be grouped into three types namely lexical, pragmatic and hyperbole.

Feature Category	Author	Method used
Lexical	Kreuz et al.[1]	The lexical feature plays a vital role to recognize irony and sarcasm in text
	Kreuz along with Caucci[2]	Used the lexical and syntactic features to recognize sarcasm in text and also discussed the role of different lexical factors, such as interjection and punctuation symbols.
	Davidov et al[3]	Used semi supervised approach to detect sarcasm in tweets and Amazon product reviews and they used lexical features namely, pattern based and punctuation based to build a weighted K- nearest neighbor classification model to perform sarcasm detection.
	Utsumi et al.[4]	Extreme adjective and adverbs often provide an implicit way to display negative attitudes

Revised Manuscript Received on March 30, 2020.
* Correspondence Author
S Maheswari*, Department of Computer Science, Bishop Heber College, Thiruchirappalli, India. Email: smaheswaribhc@gmail.com
Dr. K. Arthi, Department of Computer Science, Government Arts and Science College, Coimbatore, India. Email: dr.arthisekar@gmail.com

	Gonzlez-Ibenz et.al[5]	plored numerous lexical features derived from LWIC[19] and WordNet affect[20] to classify sarcasm
	. Riloff et al.[6]	Used a fine built lexicon based approach to detect sarcasm based on an assumption that sarcastic tweets are a contrast between a positive sentiment and a negative situation. For lexicon generation, they used uni gram, bi gram and tri gram features.
Pragmatic	Kreuz et al[1]	Used to identify sarcasm in textual data.
	Gonzlez-Ibnez et al[5]	Used emoticons, smiles and replies to identify sarcasm text.
	Tayal et al.[7]	Used to detect sarcastic phrases in political criticism .
	Rajadesingan et al[8]	Besides following the systematic way of finding fun on Twitter, it also addresses the psychological and behavioral concepts of current and old
Hyperpole	Lunando et al.[9]	To identify sarcasm words the following intensifiers were used aha,bah,nah,wah, wow, yay,uh, etc .
	Liebrecht et al.[10]	Finding the word sarcasm using hyperbole terms is much easier than finding the sarcasm word without it
	Filatova et al.[11]	A word or a sentence is not enough to detect a sarcastic sentence in a document.
	Thungamthiri et al.[12]	To identified an indirect contradiction between sentiment and situation using

The characteristic plays an important role in finding sarcasm sentences. In the literature review, researchers have used a variety of features. Researchers have used a variety of domains like tweets, product reviews, online discussion forum, Google books and website comments in their study of sarcasm detection. There are five types of sarcasm taking place in the text. They are

i) Confrontation occurs when the author likes or dislikes a fact

ii) Conflict occur when the situation is changed

iii) Being a sentence contrary to reality

iv) When hashing at the end of a speech

v) When a text difference of opinion with the reality about any event such as sports, festival, birthday etc.

The text as a sarcasm can be recognized by two classification approaches. They are machine learning approach and corpus based approach. Supervised and semi supervised approaches are used by machine learning approach. Fully labeled training set is required for supervised approach, while, part of training set is needed for semi-supervised approach. Corpus based classification falls short of the need to detect sarcastic comments which are dependent on the context used in the particular environment.

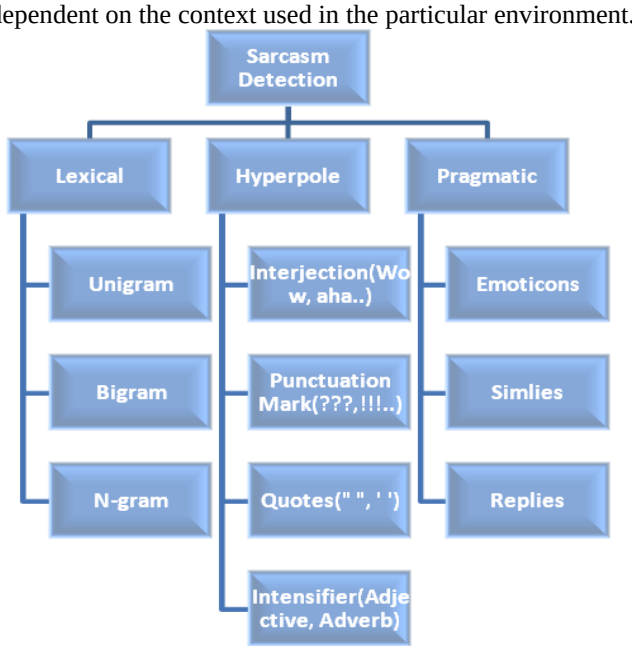

Figure 1. The features used for detecting Sarcasm in Text

In the following way the sarcasm can be occur in a dataset:

T1 : Disparity between positive situation and negative sentiment

T2 : Disparity between negative situation and positive sentiment

T3 : Reality exclusion – text contradicting a fact

T4 : resembling and aversion Prediction- behavior based

T5 : Tweets contradicting facts about event

The types of features used for detecting sarcastic sentence are listed in the above figure.

The types of domains used as follows:

D1 : Tweets Twitter

D2 : Online Product Reviews

D3 : Website comments

D4 : Google Books

D5 : Online Discussion forum

From the review most of the researches were detected the sarcastic phrase based on machine learning approach, types of sarcastic Tweets on twitter is based on type (sarcasm as a contradiction between positive sentiment and negative situation), feature list is extracted based on lexical and pragmatic, and domain used is twitter data.

III. PROPOSED METHODOLOGY

In this section, the design of the newly proposed algorithm is stated. In this paper the author has proposed algorithm to detect various types of sarcastic reviews that occur in review dataset. This proposed algorithm has been implemented in Python language. The proposed methodology has three major stages: 1) Data Acquisition, 2) Noise Reduction and 3) Sarcasm Detection.

1) Data Acquisition

Data preprocessing is the most compelling step of the algorithm proposed. The proposed dataset is 2000 reviews, which have common reviews with sarcastic and non-sarcastic reviews. This serves as input to noise reduction module. For this purpose we used two reviews datasets, namely a) kindle and b) Employee. These two dataset is publically available at Kaggle. These two datasets are stored in two separate MS-Excel files.

2) Noise Reduction

After completion of data acquisition the review data should be ready for the noise reduction. The review data contains some unwanted text. With this unwanted data the author cannot classified the sentence. So the author has to remove the unwanted text. Hence the following are the major steps used for removing unwanted text and prepare the data for the next.

1) Word Tokenization
2) Slang Correction
3) POS tagging
4) Stemming

In tokenization, the review data are split into words and remove the stop words like the punctuation, prepositions and pronouns etc. After tokenization and stop word removal the words are entering to check for slang words. If the word is in slang dictionary then that slang word is converted into equivalent normal word. After the slang correction the text are tagged by POS tag. This tagging method take place most important in classification. If there is a lot of an adjective in a sentence then that sentence is definitely a sarcastic sentence. This can be easily handled by POS tagging. After tagging, the each text is entered into stemming to find its root word.

3) Sarcasm Detection

This is a final stage in proposed method. After stemming that the text is checked for finding its subjectivity. If the sentence is subjective then the stemmed words find its polarity. Subjective is identified by using TextBlob in NLTK. The sarcasm detection in the proposed model is done using some lexicon. For sarcasm detection some features are used for detecting sarcastic reviews. So the author prepared lexicon for interjection and intensifiers with polarity value are created. The review sentence can be classified as sarcastic, only by satisfying the following rules:

1. The sentence must begin with interjection word and right way follows the adverb.
2. the review must begin with interjection word and enclose adverb and immediately followed by adjective or vice versa
3. The review sentence must begin with interjection word and hold either structure first adjective followed by a noun or first adverb followed by verb.
4. Having contradiction between negative sentiment and positive situation

The following algorithm is to detect sarcastic and non-sarcastic reviews and avoid detecting false polarity. The figure 2 clearly explains the frame work of this proposed algorithm.

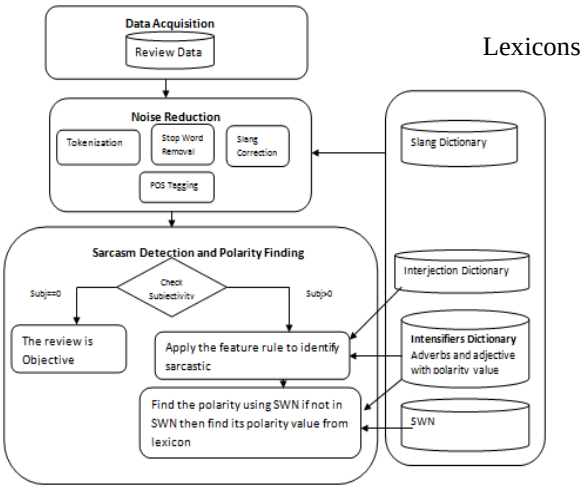

Figure 2: Proposed SRRB System

Algorithm:

Input: *Product Reviews*

Output: *Detect sarcastic, non sarcastic sentences and find polarity*

Notation: *C: ReviewCorpus, PR: ProductReview, TknS: list of tokens, TW: TokenWord, SlgF: Slang File, TWS_TAG: List of POS Tag of a Review Text, P_adjfile: positive adjective file, N_adjfile: negative adjective file, N_advfile: negative adverb file, Tot_Pos_Score: total positive score,Tot_Neg_Score:total negative score, Ftag: first tag,IMEtag: Immediate Tag, TTG: individual token from TWS_TAG*

Initialization:

```
Inter=0,NEG_SARCAS=0,POS_SARCAS=0,Tot_Pos_Score
=0,Tot_Neg_Score=0
 for PR in C do
 if find_Subjectivity(PR)!=0 then
      TknS=Find_tokenize(PR)
      TWS=normalize(TknS)
      for TW in TWS do
        if TW in SlgF then
          TW ← equivalent_slang _TEXT(TW)
      TWS_TAG ←Find_POS(TWS)
      Ftag ←TWS_TAG[0]
      IMEtag ←TWS_TAG[1]
    if Ftag in InjecFile and IMEtag in
  P_adjfile||P_advfile then
      Tot_Neg_Score ←Tot_Neg_Score+1
          Continue
    elif check Ftag in Interjection and IMEtag in
  N_adjfile ||N_advfile then
          Tot_Pos_Score ←Tot_Pos_Score+1
              continue
        for TTG in
     TWS_TAG do
```

```
        if  TTG in InjecFile then
            Injec ←1
            Continue
        if   Injec==1 and TTG in P_adjfile   then
            if  TWS_TAG[index(TTG)+1]  in P_advfile then
                Tot_Neg_Score ←Tot_Neg_Score+1
                Review is Negative sarcasm
        elif  Injec==1 and TTG in P_advfile   then
            if TWS_TAG[index(TTG)+1] in P_adjfile then
                Tot_Neg_Score ←Tot_Neg_Score+1
                Review is Negative sarcasm
        elif   Injec==1 and TTG in N_adjfile   then
            If  TWS_TAG[index(TTG)+1]  in N_advfile then
                  Tot_Pos_Score ←Tot_Pos_Score+1
                Review is Positive sarcasm
                                        Continue
        elif Injec==1 and TTG in N_advfile   then
            if TWS_TAG[index(TTG)+1] in N_adjfile then
                  Tot_Pos_Score ←Tot_Pos_Score+1
                Review is Positive sarcasm
        Elif  TTG in P_adjfile||P_advfile then
            Pindex ←index(TTG)
        Elif TTG in N_adjfile||N_advfile then
            Nindex ←index(TTG)
    End for
  If Injec!=1 and Pindex>Nindex then
        Tot_Pos_Score ←Tot_Pos_Score+1
        Review is Positive sarcasm
    Elif Injec!=1 and Pindex<Nindex then
        Tot_Neg_Score ←Tot_Neg_Score+1
        Review is Negative sarcasm
    Else
        Find Polarity using SWN and find total Positive and
negative Score
  End for
  End for
```

IV. EXPERIMENTS AND RESULTS

The proposed model is implemented in python and run in windows. The name of the proposed algorithm is Sarcasm Revealing Rule Based (SRRB) algorithm. The proposed algorithm SRRB applied to the two different dataset. Table 1 shows the results of testing with dataset1, dataset2 and its average. In both the dataset, the author first finds out the polarity of the review text without considering sarcastic. Secondly the author finds out the polarity of the reviews with considering sarcastic by applying the proposed algorithm. These details are clearly defined in the Table 2. Definitely the result of metrics value increased due to the new proposed algorithm. Here the author takes 2000 review sentences from each dataset.

The following well-known metrics were used for evaluation of the sarcasm detection task:

1) Precision: Also called Positive predictive value The ratio of correct positive predictions to the total predicted positives. This is represented in Eq.(1).

$$Pr = \frac{TP}{TP + FP} \qquad (1)$$

Where, TP: True Positive, TN: True Negative, FP: False Positive and FN: False Negative. True positives indicate the number of positive files which are correctly identified as positive. Similarly True negatives indicate the number of negative files which are correctly classified as negative. False positives denote the number of negative files which are misclassified as positive. False negatives indicate the number of positive files that are misclassified as negative.

2) Recall: Recall, Re, is defined as the number of True Positives divided by the total number of elements that actually belong to the positive class (the sum of True Positives and False Negatives). This is represented in Eq.(2).

$$Re = \frac{TP}{TP + FN} \qquad (2)$$

3) F1- Measure: This is the harmonic mean of Precision and Recall. F1 score is calculated as shown in Eq.(3).

$$F_1 = 2 \times \frac{Pos\ Pr \times Pos\ Re}{Pos\ Pr + Pos\ Re} \qquad (3)$$

4) Accuracy: The accuracy can be defined as the percentage of correctly classified instances (TP + TN)/(TP + TN + FP + FN). where TP, FN, FP and TN represent the number of true positives, false negatives, false positives and true negatives, respectively. It is calculated by the equation shown in Eq.(4)

$$Accuracy = \frac{(TP + TN)}{TP + TN + FP + FN} \qquad (4)$$

The author made the comparison with the help of various statistical parameters such as precision, recall, F1_measure and accuracy. The experimental result of the proposed algorithm attained significant accuracy as SRRB attains .87 – precision, .96 – recall, .91 - F1-measures and 86.7%. accuracy. The following Table 2 shows the polarity of the review data without using SRRB and with using SRRB.

Table 2. Precision, Recall , F-Score and Accuracy of the polarity of the review data without using SRRB and with using SRRB

Metrics	DataSet1	
	Without SRRB	**With SRRB**
Precision	0.64	0.87
Recall	0.75	0.96
F-measure	0.69	0.91
Accuracy	0.75	0.87
	DataSet2	
	Without SRRB	**With SRRB**
Precision	0.64	0.86

Recall	0.74	0.95
F-measure	0.68	0.9
Accuracy	0.74	0.86
	Average	
	Without SRRB	**With SRRB**
Precision	0.64	0.87
Recall	0.75	0.96
F-measure	0.69	0.91
Accuracy	0.74	0.87

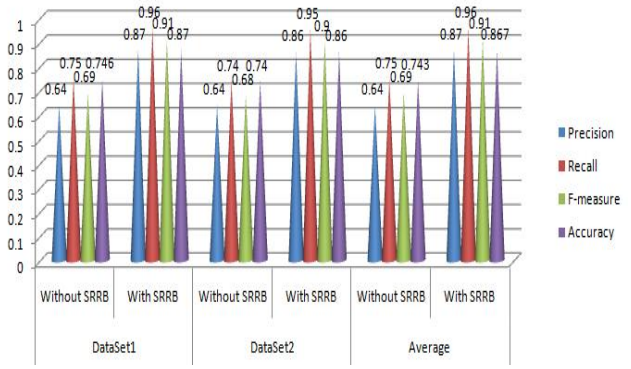

Figure 3 Precision, Recall , F-Score and Accuracy of the polarity of the review data without using SRRB and with using SRRB

V. CONCLUSION

In this paper, a way of finding sarcastic sentence has increased the polarity of sentiment analysis. The researchers used many different algorithms for classifying the sarcastic and non sarcastic review sentences. All these are clearly stated in section II. The use of the SRRB algorithm in this paper has increased the efficiency of polarity detection by identifying sarcastic review sentences. In future, we will try to achieve the better performance of this model in terms of accuracy, precision, recall and F-Measure.

REFERENCES

1. R. J. Kreuz and R. M. Roberts, "Two cues for verbal irony: Hyperbole and the ironic tone of voice," Metaphor and symbol, vol. 10, no. 1, pp. 21–31, 1995.
2. R. J. Kreuz and G. M. Caucci, "Lexical influences on the perception of sarcasm," in Proceedings of the Workshop on computational approaches to Figurative Language, ACL, pp. 1–4, 2007.
3. O. Tsur, D. Davidov, and A. Rappoport, "Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews.," in ICWSM, pp. 162–169, 2010.
4. A. Utsumi, "Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony," Journal of Pragmatics, vol. 32, no. 12, pp. 1777–1806, 2000.
5. R. Gonzalez-Ib ´ anez, S. Muresan, and N. Wacholder, "Identifying sar- ´ casm in twitter: a closer look," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-vol 2, pp. 581–586, 2011.
6. E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation.," in EMNLP, pp. 704–714, 2013.
7. D. Tayal, S. Yadav, K. Gupta, B. Rajput, and K. Kumari, "Polarity detection of sarcastic political tweets," in Computing for Sustainable Global Development (INDIACom), 2014 International Conference on IEEE, pp. 625–628, 2014.
8. A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm detection on twitter: A behavioral modeling approach," in Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pp. 97–106, 2015.
9. E. Lunando and A. Purwarianti, "Indonesian social media sentiment analysis with sarcasm detection," in Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on IEEE, pp. 195–198, 2013.
10. C. Liebrecht, F. Kunneman, and A. van den Bosch, "The perfect solution for detecting sarcasm in tweets# not," Association for Computational Linguistics, pp. 29–37, 2013
11. E. Filatova, "Irony and sarcasm: Corpus generation and analysis using crowdsourcing.," in LREC, pp. 392–398, 2012.
12. P. Tungthamthiti, K. Shirai, and M. Mohd, "Recognition of sarcasm in tweets based on concept level sentiment analysis and supervised learning approaches," pp. 404–413, 2014.
13. R. Cynthia Monica Priya, Dr. J.G.R. Sathiaseelan," A Rule Based Stemmer", International Journal of Engineering and Advanced Technology (IJEAT),ISSN: 2249 – 8958, Volume-X, Issue-X,2019

AUTHORS PROFILE

Mrs. S. Maheswari, completed her MCA degree in 2000. She completed her Post Graduate Diploma in Computer Application. She completed her M.Phil degree at Bishop Heber College in 2013. She worked as an IT Trainer at Tiruchirappalli Regional Engineering College Science and Technology Entrepreneurs Park for 8 years. Currently she is working as an Assistant Professor of Computer Science department at Bishop Heber College., Trichy. She is pursuing Ph.D at Bharathiar University, Coimbatore. She has Cleared SET in 2018. She has 15 years of experience in computer field. She has cleared Microsoft Database Administration Certificate exam. She has presented papers in International Conferences and has published papers in reputed journals. Her area of specialization is Data Mining and in particular Web Mining.

Dr. K. Arthi, MCA.,M.Phil.,Ph.D is working as an Assistant Professor in Department of Computer Applications, Government Arts College, Coimbatore. Her area of specialization is Soft computing, Evolutionary Algorithms, Fuzzy cognitive maps. She has more than 19 years of teaching experience. She got her Ph.D research degree in Anna University, Chennai. She is the life member of ISTE. She has published many papers. She has successfully guided a number of M.Phil and Ph.D scholars. At present, she is guiding both M.Phil and Ph.D scholars in Bharathiar University and Karpagam University. Her research area include, Data Mining, Soft Computing.