International Journal of Recent Engineering Research and Development (IJRERD)
ISSN: 2455-8761
www.ijrerd.com || Volume 03 – Issue 02 || February 2018 || PP. 46-54

Sarcasm Detection in Reddit

B. Lakshmanan[1], A. Anjana[2]
1: Assistant Professor (Senior Grade), Mepco Schlenk Engineering College, Sivakasi, India
2: Associate Engineer, Quantium Analytics Pvt Limited, Hyderabad, India

Abstract: Sarcasm refers to the use of satirical or ironic language to convey a message. It is usually used in social networks such as Reddit, Twitter etc. The identification of sarcasm can improve the efficiency of sentiment analysis. Sentiment analysis refers to analyzing the attitude of people towards a particular topic or scenario. Our proposed method uses a supervised approach to learn the sarcastic patterns for classification. The parts of speech tags of the posts are identified and the most frequent patterns are determined. In addition to this, multiple features are extracted and their weight is studied. The features are of four types, namely syntactic, semantic, sentiment and pattern related features. The proposed method evaluated on Reddit dataset of about 10,000 posts comprising 5,000 sarcastic and 5,000 non-sarcastic.
Index Terms: patterns, frequency matrix, resemblance degree.

I. INTRODUCTION

Sarcasm is employed to convey a meaning different than the literal one, usually in satirical context. It is complex and a bit difficult to comprehend since the actual message in the text has to be interpreted by the user. Sarcasm is utilized for humor as well as criticism of ideas, people or events. It is widely prevalent in micro blogging platforms and internet forums.

Sentiment analysis refers to the understating and analysis of the different opinion of users. It is usually done by identifying the context of the text i.e. positive, negative, neutral etc. However, sarcasm actually conveys a different meaning than the actual text. The presence of sarcasm can alter the polarity of the sentiment analysis. Hence, the detection of sarcasm can enhance and refine the existing system for sentiment analysis.

Much work has been done on detecting sarcasm in twitter owing to its maximum tweet length of 140. In our experiment we want to perform sarcasm detection in text in which there is no limit for the maximum characters. The text can be a sentence, a word or even a paragraph. Hence, we work on the reddit dataset.

Reddit is an online discussion platform, where the community members can post information regarding news, politics, hobbies etc and any other areas of interest. The areas of interests are categorized as subreddits (/news, /politics etc) .Our test data is from the subreddit sarcasm where the members post sarcastic comments which are denoted by appending a /s to the text.

To detect the sarcastic posts, the frequently occurring sentence patterns are detected along with its syntactic, semantic and sentiment based features.

II. RELATED WORK

A lot of work has been done on sarcasm detection in recent years. However, their difference lies in the features considered and the type of classification.

Davidov et al (2010) used a semi-supervised method for sarcasm identification on both twitter and Amazon datasets. They had better accuracy with Amazon compared to that of twitter.

Barbieri et al (2014) considered sarcasm as a figurative language and compares it with text from four main topics namely, Education, Humor, Irony, Newspaper, and Politics. However, their model does not include patterns of words as features.

Riloff et al (2013) used a bootstrapping algorithm that automatically learns the positive and negative words to detect sarcasm in tweets.

Li Ding et al (2014) employed adaptive recursive neural networks to detect sarcasm in twitter data. However, the optimization is unstable in this case due to different values of the hyper-parameter β

Mondher Bouazizi and Tomoaki Otsuki (2016) detected sarcasm in tweets based on their patterns and four different set of features. They used different classifiers and determined their performance. However, the pattern length was pre-determined due to the text length in tweets.

Fersini et al (2015) combined several independent classifiers to achieve better performance than a single classifier. They have worked on sarcasm as well as irony. However, differentiating between them is a complicated task.

Table 1: Summary of different algorithms for sarcasm detection

Authors who proposed the algorithm	**Algorithm for analysis**	**Accuracy result**	**Difficulties with this method**
Li Dong et al.	Convert Dependency Tree and Back propagation	66.3%	The different value of hyper-parameter β leads to unstable optimization
Ellen Riloff et al.	Bootstrapping algorithm	Not specified	Identifies the sarcasm by using specific syntactic structures
Dmitry Davidove et al.	SASI -semi supervised sarcasm identification algorithm	Twitter -90.6 Amazon- 94.7	Only covers limited subset of classic patterns
Francesco Barbieri et al.	Supervised machine learning methods	Not specified	Model detecting irony, with some extra features struggles to determine the sarcastic tweets.
Modher Bousazizi, Tomoaki Otsuki	Supervised pattern Matching	83.1%	Accuracy is based on the training dataset used

Different classifiers such as SVM, Naïve ayes, Maximum entropy were used by BPang et al (2002) on movie reviews. But, they had difficulties identifying whether the reviews were on topic.

Rajadesingan et al (2015) introduced behavioral modeling of sarcasm using the current and past tweets of a user.

.

III. PROPOSED SYSTEM

Preprocessing

The URLs like http://www.google.com are removed from the posts during pre-processing. These URLs usually refer to sarcastic images and gifs. Hence, they are removed. As mentioned earlier all the posts in sarcasm subreddit have '/s' in their text .This is done by the user to indicate that the post is sarcastic. Thus, the '/s' is removed in the pre-processing stage.

For e.g.

"Ah! yet another marriage that is destined to last an eternity/s."

After processing,

" Ah! yet another marriage that is destined to last an eternity."

Lexical auxiliary is a helping verb, i.e., auxiliary verbs help the main verb. When an auxiliary verb exists, there is a verb phrase. These verb phrases usually are written in their short forms (Eg. can't, won't etc). The shortened verb phrases are replaced by their full forms to retain the actual sentence pattern. It also normalizes the text before counting the punctuation related features and tagging the sentence. The auxiliaries are initially converted to lowercase and then modified. The auxiliaries such as aren't, don't, haven't are replaced by their counter parts are not, do not, have not etc.

About 63 auxiliaries are replaced and some of the auxiliaries replaced are listed in the following table2.

For e.g.,

"Psh, are you kidding? They're obviously just lazy peons who can't grasp the value of cooking at home."

After removing auxillaries,

"psh, are you kidding? they are obviously just lazy peons who cannot grasp the value of cooking at home."

Table 2: List of auxiliaries and their replacement

Auxiliary	**Replacement**
i'm	i am
he's	he is
aren't	are not
they're	they are

we've	we have
you'd	you had
she'll	she will
they'd	they would
needn't	need not
oughtn't	ought not

Feature Extraction

Sentiment Related Features

The sentiment related features consists of count of positive and negative words in the sentence. The SentiStrength database contains the list of positive words and negative words each indicated by a value ranging from +1 to +5 and -1 to -5 respectively. By comparing the posts with the list of words, the numbers of positive and negative words are identified.

The positive word count is denoted by P and the negative word count is denoted by N, which are used in calculating the ratio of emotional words.

For e.g.,

"Yeah they are ***truly*** *a* ***threat*** *to society "*

P: 1 and N: 1.

In addition to this, the counts of highly positive and highly negative words are also identified. The lists of positive and negative words with the tags listed in table 3 are considered to be highly emotional.

Table 3: List of tags for highly emotional words

Parts of Speech	**Parts of Speech Tag**
Adjectives	"JJ", "JJR", "JJS"
Adverbs	"RB","RBR", RBS"
Verbs	"VB", "VBD", "VBG", "VBN", "VBP", "VBZ"

The highly positive and highly negative words are denoted by P_H and N_H respectively.

The emotional ratio is calculated using the formula

$$\rho(t) = \frac{(\delta.P_H + P) - (\delta.N_H + N)}{(\delta.P_H + P) + (\delta.N_H + N)} \qquad (1)$$

where,

$\rho(t)$ - the emotion ratio.

P - the number of positive words other than P_H.

N - the number of negative words other than N_H.

P_H - the number of highly positive words.

N_H -the number of highly negative words.

δ - the constant value 3.

Punctuation Related Features

The number of times the following punctuations are occurred is determined:

- Exclamation mark - !
- Question mark - ?
- Dots - .

- All capital words, words must be of length greater than 2.
- Quotes - ‘,”
- Star - *

In addition to these features, the number of words in the sentence and repetition of vowels is also counted. The vowels must occur consecutively more than twice.
Eg.
“ Ohhh... ***Nooo****... we need to trade for a better RB”*

Syntactic and Semantic Features

The uncommon words are identified by removing the common stop words from the text such as of, an, a etc. The remaining word count is calculated.
For e.g.
“ah yet another marriage that is destined to last an eternity “
After removing stop words,
“ah yet another marriage destined last eternity “
The feature count is set to 7.
The presence of common sarcastic expression such as “lol”, “haha”, “rofl” etc are identified. These are considered as one of the feature during sarcastic post identification.
For e.g.
”*Rolls away* ***lol*** *classy “*
The laughing expressions like :P, :D, ^_^ etc are used in the post to expose sarcasm. The sentences are compared with the emoticon list of SentiStrength database. This also considered while identifying the sarcasm of the post
For e.g.
“Try to be a better scanner in the future :P”
The posts are then tagged using the Part of Speech Tagger in GATE (General Architecture for Text Engineering). The tags used are from Penn Tree Bank.
For e.g.
“do not you know that any arrow in any place is an obscure reference to reddit!!”
After Tagging,
“VBP RB PRP VBP IN DT NN IN DT NN VBZ DT JJ NN TO NN.”
Based on the presence of “UH” tags, the interjection count is detected. This is taken as one of the features of sarcasm identification.
For e.g.
“ah yet another marriage that is destined to last an eternity”
Tags are,
*“****UH*** *RB DT NN WDT VBZ VBN TO VB DT NN.”*
Feature count is set to 1.
After part-of-speech tagging, the functionality of each tag is identified. The tags are broadly classified into two types based on their functionality, namely CNT and GF. In CNT tags, the context of the tag is considered, whereas in GF tags, the grammatical function of the tag is considered. The list of tags which come under CNT and GF are listed in the table 4,

Table 4:List of CNT and GF tags

CNT	CC, DT, EX, IN, JJ, JJR, JJS, PDT, POS, RB, RBR, RBS, RP, TO, VB, VBD, VBG, VBN, VBP, VBZ
GF	CD, FW, UH, LS, NN, NNS, NNP, NNPS, PRP, PRP$, MD, PB, RBR, RBS, WDT, WP, WP$, WRB, SYM

The CNT tagged words need to retain their content, hence they are replaced by their lemmatized form. For lemmatization, Helmut Schmid tree tagger is used.
For e.g.
running, runs, run => run
For the sentence,
“if we evolved from bugs then how come we still have bugs.”
The POS tags are,

"IN PRP VBD IN NNS RB WRB VB PRP RB VBP NNS ."
After replacing CNT tags,
"if PRP evolve from NNS then WRB come PRP still have NNS ."
The GF tags do not need to retain their original words. Hence, filler tags or placeholders are used in their place. The following table shows the filler tags,
For e.g.
"ah yet another marriage that is destined to last an eternity."
After replacing the GF tags,
"INTERJECTION yet another NOUN WHDETERMINER be destine to last an NOUN "
Table 5 contains the list of GF tags and their corresponding placeholder or filler tags.

Table 5: List of GF tags

POS –Tag	Expression
CD	CARDINAL
FW	FOREIGNWORD
UH, PRP, PRP$	INTERJECTION
LS	LISTMARKER
NN, NNS, NNP, NNPS	NOUN
MD	MODAL
PB, RBR, RBS	ADVERB
WDT, WP, WP$, WRB	WHDETERMINER
SYM	SYMBOL

Pattern Extraction

The pattern formed after replacing the tags with their corresponding GF, from which sub patterns of length from 2 to 14 are formed. From these patterns the most frequently used are identified. The non-sarcastic patterns are removed in order to get the actual sarcastic pattern.
For e.g.
"ah yet another marriage that is destined to last an eternity."
"INTERJECTION yet another NOUN WHDETERMINER be destine to last an NOUN"
List of sub-patterns of length 2,
{{INTERJECTION yet}, {yet another}, {another NOUN}, {NOUN WHDETERMINER}, {WHDETERMINER be}, {be destine}, {destine to}, {to last}, {last an}, {an NOUN}}
List of sub-patterns of length 3,
{{INTERJECTION yet another}, {yet another NOUN}, {another NOUN WHDETERMINER}, etc.}
List of sub-patterns of length 4
{{INTERJECTION yet another NOUN}, {yet another NOUN WHDETERMINER}, {another NOUN WHDETERMINER be}, etc}
List of sub-patterns of length 5
{{INTERJECTION yet another NOUN WHDETERMINER}, {yet another NOUN WHDETERMINER be}, etc}
Similarly all possible sub-patterns up to length 14 are determined for both sarcastic and non-sarcastic data which occurs more than twice. Then, the set of patterns which are common in both are removed from the list of sarcastic patterns.

Pattern Recognition

The feature set can be created by

$$N_F = N_L * N_s \qquad (2)$$

where,

N_F represents the feature set.

N_L represents the number of pattern lengths.

Then the possible set of sub-patterns are detected and the most frequently occurring sarcastic patterns are filtered out.

The frequency of the detected pattern for each pattern length are identified and are projected in the figure 1.

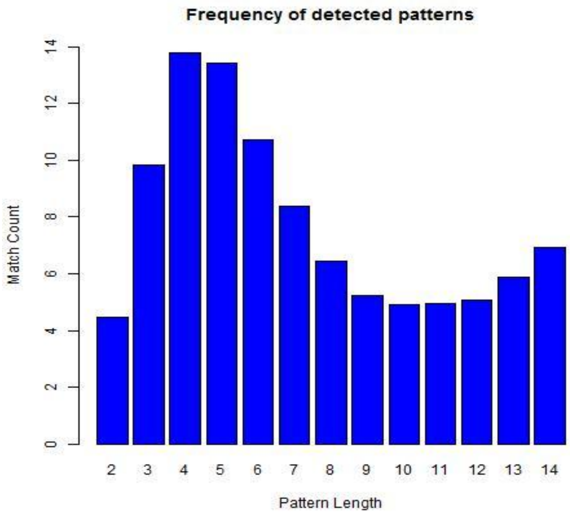

Figure 1: The frequency of the detected patterns of various lengths

From the above graph it can be noted that the maximum value is for patterns of length 4.Hence it is taken as the minimum pattern length for matching .Then the curve falls, and begins to rise again at length 11.This is because longer length patterns have very low probability of being general and will not be removed when subtracting the set of serious patterns. Hence, the pattern length 10 is considered as the maximum pattern length to be extracted.

We consider the patterns of length 4 to 10 and all the features extracted previously to train the various classifiers namely, SVM, Naïve Bayes, Random Forest and Simple CART.

We normalize the feature matrix with a range 0 to 100. By analyzing these values, we find that pattern length 4 has the maximum detected sarcastic patterns and length 10 is the steep point of the curve. After that, the detected values increases, this is because the sarcastic patterns are retained during subtraction as they are very rare.

Hence, L_{min} =4 and L_{max} =10 represent the minimal and maximal allowed length of patterns in words.

For the testing set, the resemblance degree is identified using

res(p,t) = 1, if the post vector contains the pattern as it is, in the same order,

else,

0, if the pattern does not appear in the post.

where, p -patterns , t – posts. The sum of the resemblance degree for all the patterns of each length ranging from 4 to 10 is identified. This feature is also used in training the classifier.

IV. Evaluation

Cross Validation

After getting the resemblance degree of the posts, we perform 8 fold cross validation using four different classifiers namely Naive Bayes, Random Forest, Support Vector Machine and Simple CART. The performance measures of these classifiers are listed in the table 6.

Table 6: Performance Measure of Different Classifiers

Classifiers	Precision	Recall	Accuracy
Random Forest	0.647	0.647	64.68
SVM	0.659	0.622	62.18
Naives Bayes	0.649	0.599	59.19
Simple CART	0.663	0.663	66.29

Among the classifiers, Simple CART has the highest accuracy of 66 %.The accuracy can be further improved by enriching the sarcastic patterns used to train the classifiers.

In addition to this, the accuracy of patterns based on their length is also determined using Random Forest Classifier. The accuracy values are displayed in table 7.

Table 7: Performance Measure of Different Classifiers

Pattern Length	Percentage of Accuracy
4	63.73
5	62.76
6	62.06
7	61.65
8	62.37
9	61.9
10	61.91

Test Data Evaluation

The test data considered consists of a list of 1000 posts which are both sarcastic and non-sarcastic. On evaluating the test data, we get the performance measure as shown in table 8.

Table 8: Performance Measure for Test Data

Classifiers	Precision	Recall	Accuracy
Navies Bayes	0.563	0.568	56.84
SVM	0.552	0.557	55.65
Random Forest	0.575	0.561	56.08
Simple Cart	0.56	0.549	54.89

The accuracy results of these various classifiers while performing 8 fold cross-validation are shown in figure 2

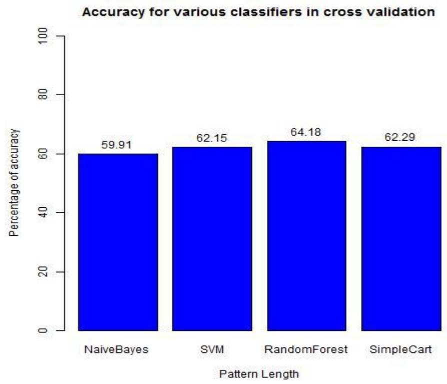

Figure 2: The accuracy results of various classifiers in cross validation

Among the classifiers Random Forest has the highest accuracy of 64 %.This can be further improved by enriching the set of sarcastic patterns.

The accuracy for each individual pattern lengths were determined and plotted as a graph shown in figure 3. The classification was performed using random forest since it had the highest accuracy among the classifiers.

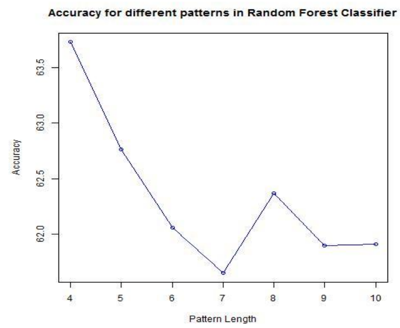

Figure 3: The accuracy results of various pattern lengths using Random Forest

Here, the pattern length of 4 has highest accuracy of 63%.The same set of classifiers are used on the test data and their performance is measured. The results are displayed in figure 4.

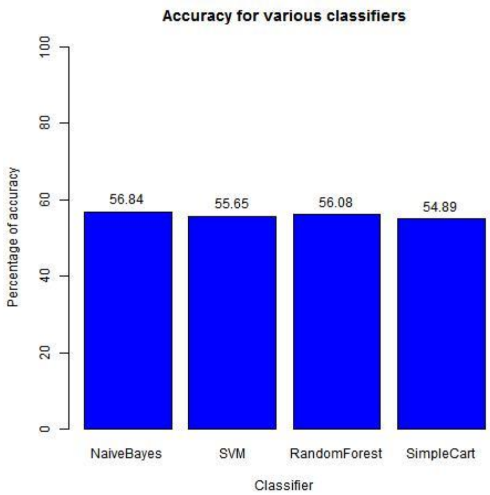

Figure 4: The accuracy results of various classifiers using test data

The best result is given by Random Forest classifier as same as during cross validation. The accuracy goes as far as to 56%.

V. CONCLUSION

The proposed method was used to detect sarcasm in the internet posts. The approach was implemented on the Reddit dataset consisting of 5000 sarcastic and non-sarcastic posts. The test data comprised of 1000 posts with equal number of sarcastic and non-sarcastic posts. The different types features are extracted from the posts and based on the similarity to the training set the posts have been classified as sarcastic or not. The features that were considered includes sentiment related features, punctuation related features, syntactic and semantic features and pattern related features .The method identifies mostly all the sarcastic post, because all the sub-patterns of the post are also verified during processing. The model considers the sub -patterns of length from 4 to 10. This makes the model to classify the posts more accurately. The accuracy of the model is comparatively good. The analysis with different classifiers has been made to compare the performance of the various classifiers for sarcasm detection.

The proposed method works on a limited dataset of about 10,000 posts comprising 5,000 sarcastic and 5,000 non-sarcastic. In future, dataset can be enriched such that more sarcastic patterns are identified and the efficiency of the model can be increased. Since the dataset has large posts, the sub-pattern length may also vary in order to get better result. After classification, a further extension can be done to identify the actual sentiment of the sarcastic post i.e. whether the sarcastic post is positive or negative.

ACKNOWLEDGMENT

First and foremost, we thank the Almighty for his abundant blessings that is showered upon our past, present, future successful endeavors.

At the appellation, we would like to extend our sincere gratitude to our reverent principal, Dr. S. Arivazhagan, M.E., Ph.D., for providing sufficient working environment such as system and library facilities.

We also thank him very much for providing us with adequate lab facilities, which enable us to complete our work.

We are grateful to our charismatic Head of the Computer Science Engineering and Department Dr. K. Muneeswaran, M.E., Ph.D., for giving us this golden opportunity to undertake a work of this nature and for his most valuable guidance given at every phase of our work.

We acknowledge with the sense of gratitude, the help given to us by all the technicians in the department.

We are very grateful to our beloved parents and friends who afforded the necessary help for us at the right time for making our work a grand success.

REFERENCES.

[1] Mondher Bouazini, Tomoaki Ohtsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter" IEEE Access, Vol.4, pp.5477-5488, August 2016.

[2] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang|, Ming Zhou, Ke Xu, "Adaptive Recursive Neural Network for Target- Dependent Sentiment Classification in Proc. Annual Meeting of Associaton of Computational Linguistics,pp. 49-54,June 2014

[3] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Si lva, Nathan Gilbert, Ruihong Huang,"Sarcasm as Contrast Between a Positive Sentiment and Negative Situation", Proc. of the 2013 Conference on Empirical Methods in NLP, pp 704-714, Oct. 2013.

[4] Dmitry Davidove,Oren Tsur, Ari Rappoport, "Semi-Supervised Recognition of Sarcastic Sentence in Twitter and Amazon", Proc. Of 14th Conference on Computational Natural Language Learning, pp. 107-116, July 2010.

[5] F. Barbieri, H. Saggion, and F. Ronzano,''Modelling sarcasm in Twitter, a novel approach,'' in Proc. WASSA, pp. 50–58, June 2014.

[6] Oren Tsur, Dmitry Davidov and Ari Rappoport," ICWSM – A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews", Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media.

[7] B. Pang, L. Lillian, and V. Shivakumar, ''Thumbs up?: Sentiment classification using machine learning techniques,'' in Proc. ACL Conf. Empirical Methods Natural Lang. Process., vol. 10, pp. 79–86 July 2002.

[8] S. K. Bharti, K. S. Babu, and S. K. Jena, ''Parsing-based sarcasm sentiment recognition in Twitter data,'' in Proc. IEEE/ACM ASONAM, pp. 1373–1380, August 2015.

[9] D. Bamman and N. A. Smith, ''Contextualized sarcasm detection on Twitter,'' in Proc. AAAI Int. Conf. Web Soc. Media (ICWSM), pp. 574–577, May 2015.

[10] A. Reyes, P. Rosso, and T. Veale, ''A multidimensional approach for detecting irony in Twitter,'' Language Resources and Evaluation, vol. 47, no. 1, pp.239–268, March 2013.