A hybrid approach for Sarcasm Detection of Social Media Data

N.Vijayalaksmi*, Dr. A.Senthilrajan**

*Lecture, Dept., of MCA, Mount Carmel College, Bangalore
**Director, Computer Centre, Karaikudi
*nvijayalaksmi123@gmail.com

***Abstract*:** In the course of the most recent couple of decades, Social networking and micro blogging websites such as twitter has allowed people to encounter in utilizing online assets. Twitter is rapidly gaining popularity as they allow the users to express and also share their opiniosns about certain related topics, which have been a mode for discussion with different communities, and messages are being posted across the world via this Social networking medium. A great deal of improvements has been observed in the field of sentimental analysis of twitter data. This project focuses mainly on sarcasm detection which is a major part of sentiment analysis of twitter data which is helpful to analyze the sarcasm in the tweets where views are miscellaneous and highly unstructured, or may be positive, negative, sarcastic, ironic or neutral in some cases. This research work borrows the ideas of utilizing different semi-supervised algorithms like Lexical Analysis with N-grams approach, Knowledge extraction, contrast approach, emoticon based approach and hyperbole approach to propose a new rule based Hybrid approach for sarcasm detection.

***Keywords*:** Hashtag Approach, Contrast Approach, Bigrams – N-grams Approach, Hyperbole Approach, Emoticon based, Hierarchical Approach

I. INTRODUCTION

The process of converting a sequence of characters into tokens is called Lexical Analysis. In a tweet, where there are 140 characters which are parsed by a lexical analyzer there are numerous tokens created. These tokens allows the machine to know, learn and predict whether the tweet is sarcastic or not with various features. The tweet characters given as input is parsed with the lexical analyzer and utilized to give tokens as our output. The probabilities of a tweet being sarcastic in the given tokens are analyzed in order to compute. This process is iterated over various tokens. Then the created tokens, coordinates with the tokens having few phrases in dictionary and later passed to the algorithm.

Twitter has many components such as "hashtag" and “mentions’ which are considered as a type of metadata components. Hashtags are the frequently used metadata component which makes it possible to get set of messages a user can search for the tweet containing “hashtags” from various groups’ messages. A hashtag are/is associated with a particular medium and in the same manner does not allow be connecting and associating with pictures or messages from various stages. It is also observed that over 90% of the users use hashtags in their tweets. Hence, analyzing tweets based on hashtags can help us determine the result of our outcomes. “Sarcasm" can be utilized as a pursuit parameter, which may help us to obtain tweets with a slight possibility of having sarcasm in them. In such cases the user explicitly tries to hint use of sarcasm in a tweet. However, some users on twitter utilize “#sarcasm” not as a random hashtag but as a part of a sentence, which tells that every tweet having #sarcasm is not considered to be sarcastic. Hence, the algorithm assures that it doesn't assume every one of tweets with “#sarcasm “as sarcastic. Rather, it identifies the use of “#sarcasm” in a tweet, as either a hashtag or as a part of the sentence. In the latter case, the tweet is considered to be something about sarcasm but it is not classified as being sarcastic.

The algorithm checks whether there exists a token “#sarcasm”. If the token is available as a part of the sentence, it cannot be declared that tweet is sarcastic. Hence, the tweet listed into Non sarcastic list.

E.g. every #sarcasm is not considered as #sarcasm. – Non Sarcastic

On the other hand if the component does not exists as a part of the tweet it’s considered to be sarcastic if there exists at least three words with meaningful sentences.

E.g. the food was really great yummyyyyy. #sarcasm– Sarcastic

The algorithm observes that #sarcasm is at the end of the sentence and hence can deduce that the user intentionally was attempting to say that the tweet is sarcastic. Many researches showed that the Tweets contain some regularly used phrases/words which hints the probability of sarcasmin a sentence. While utilizing administered learning to build the knowledge base, if a considerate number of tweets utilize a specific phrase, the observed phrase is set apart of the new trained data set assumed to have knowledge of sarcasm, analyser adds it to the knowledge base. While parsing, when it is observed that any of these words are utilized as a part of given tweet, that tweet can be assumed as sarcastic which is proved in this case study by utilising lexical approach as part of machine learning approach. Finally, this becomes our first rule in rule based approach. Hashtag based approach is very essential and initial step to the new rule based engine that we have developed. The latter rules proves that sarcasm can also be detected without “#sarcasm” tags using combinational approaches involving POS logic [4], emoticons [3] etc. There are 7 rules or phases in a rule based approach and are as follows:

1) Hash tag based approach
2) Contrast approach
3) Lexicon based approach (N-grams=Bigrams)
4) Emoticon based approach
5) Hyperbole approach
6) Hierarchical approach

7) Hybrid approach

The Contrast approach uses a logic i.e. a tweet having a positive sentiment and negative situations is considered to sarcastic and are marked as sarcastic. On the other hand Lexicon based approach consists of bag of words and trained dataset to determine sarcasm in a tweet text. Since the twitter platform allows the user to use emoji to express their sentiments, hence a rule can be built based on the given feature. Emoticon based approach is used to determine text based on a logic having a positive sentiment tweet with negative emoticons and negative sentiment tweets with positive emoticon can be assumed to be sarcastic In many instances. When the machine determines the emoticon logic it is automatically tagged as tweet and we move on to the next phase. The next phase in rule based engine is Hyperbole approach which can be considered the most convenient and important phase for sarcasm detection. It is based on a logic where we consider a unigram set of trained dataset with pragmatic markers solution is able to determine the sarcasm in a text. Unigram dataset contains set of intensifiers ('wow','Awesome','nottttt') which expresses the emotions as exaggerations (as intensifiers') of a user. And the pragmatic markers are a way expressing words by representing them with various font styles (as [WRITING IN SQUARE BRACKETS] or "WRITING IN CAPITAL LETTERS" or use of "Exclamatory!!!!!!!!! FULLSTOPS…….. /delimiters" etc...). Hence a proper solution is built using the above logic and we were able to determine sarcasm in texts more often. 6th rule of rule based approach is hierarchical approach which determines text as sarcastic through a sequential logic which takes into considerations.

Finally comes the main phase i.e. hybrid approach it's a combinational logic of all the given approaches as shown in fig 1,which gives a high precise rate of sarcasm in a text and it was shown that it gave wonderful results compared to all the other approaches. Hence we were able to determine 75% of tweets to be sarcastic and 20% non-sarcastic and rest of the tweets were ignored as they were not marked under any rule based phases (which suggests that it might have been an empty tweet (after filtering process)).

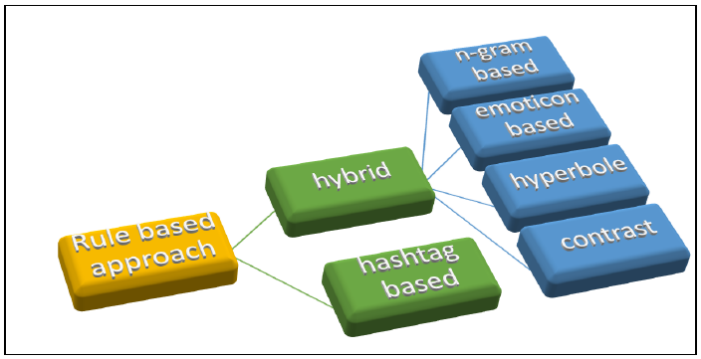

Figure 1: Rule based approach

II. RELATED WORK

There have been many efforts seen in social media analysis on trying to understand the community cohesion, opinion analysis, marketing strategies, etc. There have been seen many different proposals made on-the-fly for social media sentiment analysis, such as Prometheus in [7] and Truthy in [8]. A "P2P" service called "Prometheus" is used, that uses multiple sources to collect and manage social information in order to apply social inference functions, while on the other hand "Truthy" is a web service used for tracking political memes in Twitter and helps to detect various misinformed content in the various contexts such as of U.S. political elections etc.

Feature based classification approach or FBCA in [9] is considered the most suitable approach for sentiment analysis on twitter data, mainly reason being that the Keyword based approach cannot be ported to other languages which is a big disadvantage as tweets are tweeted in a multilingual form expressed as one, hence it was shown that FBCA got more accurate and adequate results compare to keywords based approach.

For over a large corpus of data, the features were represented through a "bag of words" model which generated a sparse matrix by utilizing a weighing scheme, this was achieved by limiting the results to a minimum of three characters word length. The case study [17] showed that by applying tokenization, removing stop word and by converting the words to lower case functions only individual word features were obtained. To overcome the following flaw a new feature was then introduce that represented as "bag of words" model which utilized the weighing scheme to generate a sparse matrix, by limiting the result to a minimum word length of three characters (words < 3). Later by evaluating this method, we found that the accurate result depends on several factors and is beyond the scope of this work.

Various lexicon-based sentiment analysis and classification methods are widely used in case studies [12], [13] and [14]. Gonçalves et al. conducted many researches to find out which method worked most effectively for sentimental analysis by comparing various existing lexicon-based methods. A broad overview of all existing works is shown in [10]. Our work is related to but different from the work in[11] which relates to the problem of generating feature-based summaries on reviews of a customer.

In [16], Go et al. was responsible for the proposal of the most advanced state of art researches in the field of twitter sentiment analysis field. Which explored the possibility of training data with a combinational logic of both Part of speech Tags (POS) having valence values with various n-grams features for the feature result set? A new parsing model was developed as a framework to implement combinational logic of different N-grams such as unigrams or bigrams, unigrams and bigrams, and unigrams and POS tags to detect the sentiments of tweets. Emoticons such as ":-), :-/, :-(, ;-p" which determined the positivity and negativity emotion in text were used to collect training data using twitter API.

The most effective and widely used method for sentimental analysis is based on supervised learning model because it is essential for an information to be known by the developer rather than letting the machine alone learn and deduce, for which the user has no knowledge on what basis the algorithm analyses. Conditional Random Fields (CRF) is used in [8] to extract targets by learning the patterns. A set of domain independent features such as syntactic dependency and parts of speech tags are used to

train CRF from multiple domains. It was observed that other learning methods can also be used.
A contextualized approach was shown in [18] for sarcasm detection. This made use of rigorous analysis of twitter data for sarcasm detection. This paper followed tenfold featured rule which proved to be some of the advanced technique for context approach. Some of the features were word unigrams, POS, Brown cluster Unigrams, pronunciation features and so on.

III. DATA

Recent works showed that, Sarcasm detection on Twitter found very less percentage of human annotators who were able to judge the sarcasm of others' tweets. Later it was seen that the recent research exploits the users' self-declarations feature for hinting sarcasm in the form of a metadata component #sarcasm or #sarcastic tags in their own tweet. We were able to find out the design choice which was able to capture common properties of sarcasm expressed without an explicit hashtag, but does not mean its' always sarcastic.

The case study exploits the user's self-reporting of sarcasm feature and follow the same methodology, by identifying all the tweets mentioning #sarcasm or #sarcastic in the #tag tweets from 2013–16 regardless of who the author is, and we were able to collect the most recent 2000 tweets of those authors. The tweets are collected from Node-xl. First we preprocess the given tweets by checking for all unwanted dependencies such as misspelled words, remove retweets, @mentions, remove hyperlinks, links, email and separate the words from the pragmatic markers and produce a fresh set of tweets to python machine to yield highly positive and precise results.
This process is entirely done by the parser.

This yields a sarcastic training set of 2500 tweets, for non-sarcastic data, we select an equal number of tweets from users over the same time period who have not mentioned #sarcasm or #sarcastic in their messages. The total dataset is evenly balanced at 3000 tweets. Since the hashtags #sarcasm and #sarcastic and #not are used to define the sarcastic examples, we remove those tags from all tweets for the prediction task.

IV. IMPLEMENTATION

4.1. Rule 1: Hashtag Approach

Hashtag based approach:
We have created a unique dataset dictionary for hashtags and represented them as unigram set of words. The most widely used unigram hashtags for sarcastic comments are #Not, #sarcasm, #sarcastic etc.

A program logic as shown in fig. 2 is developed to identify these unique tags and make a comparison test to benefit our research for accurate results and classification of tweet data as sarcastic or not.

Input: Processed'Tweets'
Logic:
From *tweetdict* import *tweets:*
For each *tweet* in *tweets:*
For *data* in *tweets*:
If *data==#Sarcasm-Data:*
Append *'Sarcastic'* to [Output-list]
Else:
Append 'Non-Sarcastic' to [Output-list]
For *tweet* in *tweets:*
For*listdata* in *Output-list:*
Newlst=Join (*tweet* with *list-data)*
Output: [*Newlist*] **of tweets**

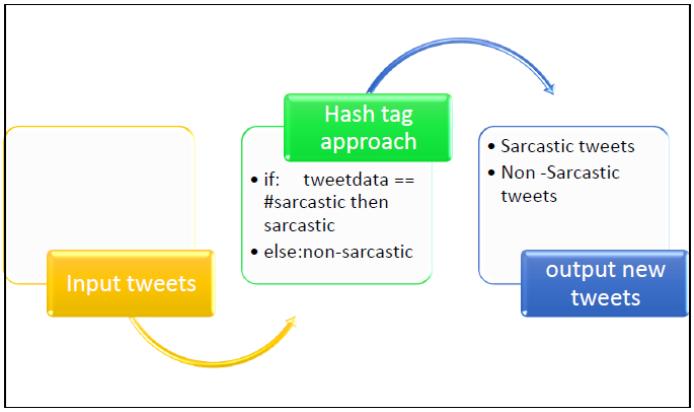

Figure 2: Hash tag Approach

4.2. Rule 2: Contrast Approach

Rule2:
Previous works have proved that parts of speech information is amongst the most informative and productive approach for this task.

We have applied the POS tagger and respective valence value from Warner et al. (2013) Dataset which includes features based on the absolute count and ratio of each tweet, along with the "lexical density" of the tweet, which models the ratio of nouns, verbs, adjectives and adverbs to all words a baseline capacitor (General Dictionary) is developed and implemented for productive analysis and classified data as shown in fig 3
Input: Processed 'Tweets'
Logic:
From *tweet-dict* import *tweet*
From *Pos-dictionary* import *All-pos*
For each *tweet* in *tweets:*
For *data* in *tweet:*
#Contrasting Approach
If *data* is foundin*Positive-verb-box* and *negative-situation:*
Append *'Sarcastic'* to [*Output list]:*
Else:
Append *'Non-Sarcastic'* to [*Output list*]
For *tweet* in *tweets*:
For *listdata* in *list:*
NewlistJoin(*tweet* with *Output-List)*
Output: *[Newlist]* of tweets

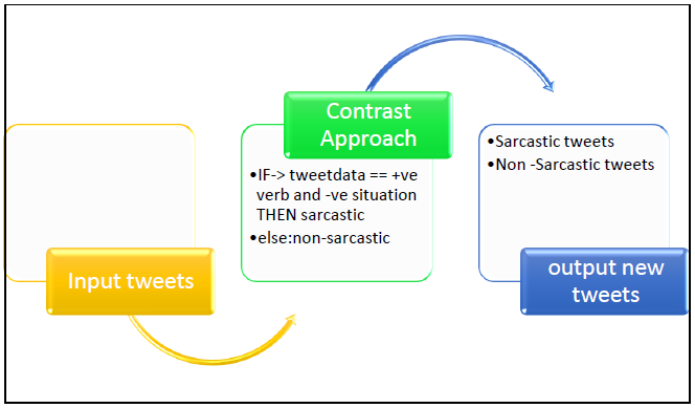

Figure 3: Contrast Approach

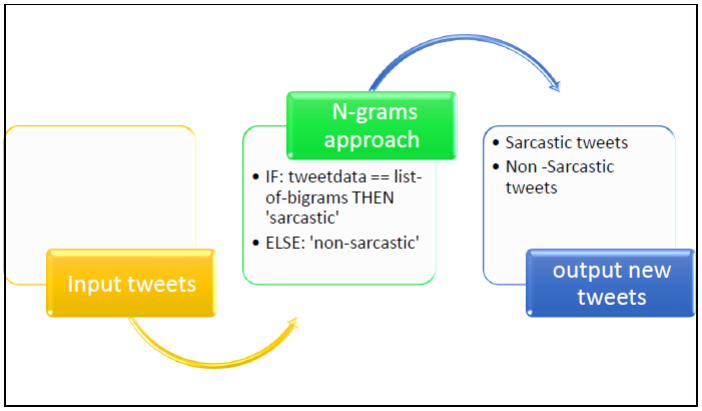

Figure 4: N-gram Approach

4.3. Rule 3: Bigrams – N-grams Approach

N-gram Approach:

Patterns can be created by concatenating adjacent tokens into n-grams where n is the max value on which the all possible combinations can be made for a single pattern.In other words here n is equal to one hence called as unigram. An example of how the n-grams are constructed is explained with the example sentence: "jack likes his new backpack", by using n-grams it may be possible to capture how a word N-grams sequence:

N-gram Sequence Length

1-gram:"jill","likes","her","new","shades" 5

2-gram:"jill likes", "likesher", "hernew", "newshades" 4

3-gram:"jill likes her", "likes her new", "her new shades" 3.

All tend to appear in text with respect to other words. Usually n-grams are never longer than n = 3. Greater values are likely to create too many complex patterns that rarely match.

In fig 4 the N-gram process is being checked for all bigram values and marked as sarcastic or non-sarcastic.

Input: Processed 'Tweets'

Logic:

From *tweet-dict* import *tweet*

From *Pos-dictionary* import *All-Pos*

From *Bi-gram* import *ngram-dict*

For each *tweet* in *tweets:*

For *data* in *tweet:*

#Ngram Algorithm

If *Bigram-data* is found in *bigram-dict*:

Append *'Sarcastic'* to [*Output-list*]:

Else:

Append *'Non-Sarcastic'* to [*Output-list*]:

For *tweet* in *tweets*:

For *listdata* in *list:*

Newlist Join (*tweet* with *Output-List)*

Output*: [Newlist]* of tweets

4.4 Rule 4: Hyperbole Approach

Hyperbole Approach:

Kreuz and Roberts 1995 theoretical work has stressed the importance of hyperbole for sarcasm, we have implemented indicator logic for whether the tweet contains a word in a list of intensifiers (so, too, very, really), stretched words (righttttt), capitalized words ([HELLO]), punctuations and so on. Hence giving us a more stable classifier hence covering most of the uncovered area for text classification as shown fig 5 and analysis.

Input: Processed 'Tweets'

Logic:

From *tweets* import *tweet:*

From *unigram* import *ngram_dict:*

For each *tweet* in *tweets*:

for*data*in *tweets:*

#Ngram Algorithm

If *unigram-data* is found in {*unigram-dict*}:

Increment count1 by 1

If *unigram-data* is found in {*punctuation-dict*}:

Increment count2 by 1

If *count1, count2* [Greater than] *fixedvalue*:

Append *'Sarcastic'* to [*Output-list*]

Else:

Append *'Non-Sarcastic'* to [*Output-list*]:

For *tweet* in *tweets*:

For *listdata* in *list:*

Newlist Join (*tweet* with *Output-List)*

Output*: [Newlist]* of tweets

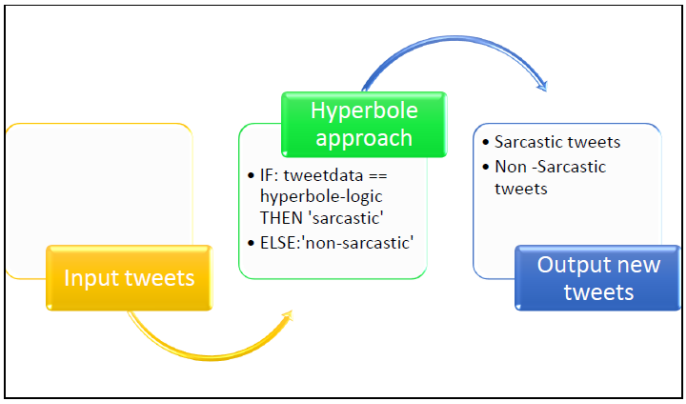

Figure 5: Hyperbole Approach

4.5. Rule 5: Emoticon based

Emoticon Based:

We have developed a dictionary containing the Unicode values for all available emoticons and we use the positive-verb dictionary from "posdict" and assuming that a tweet having a positive sentiment followed by a negative sentiment emoticon is said to be sarcastic based on this theory we are mark and divide the tweets as sarcastic and non-sarcastic as shown in fig 6

The most commonly used emoticon to comment a sarcastic message in twitter or any social media is upside-down emoticon and it's been logically proved.

Input: Processed 'Tweets'

Logic:

From *tweets* import *tweet*
From *posdata* import *all_dict*
From *emoji* import *emoji_dict*
For each *tweet* in *tweets:*
For *data* in *tweets:*
#postagging, emoticonTagging
If *unigram-data* is found in all_dict:
Increment count1 by 1
If unigram-data is found in emoticon-dict
Increment count2 by 1
If count1 =positive and count2 = negative
Append *'Sarcastic'* to [*Output-list*]:
Else:
Append *'Non-Sarcastic'* to [*Output-list*]:
For *tweet* in *tweets*:
For *listdata* in *list:*
Newlist Join (*tweet* with *Output-List)*
Output*: [Newlist]* of tweets

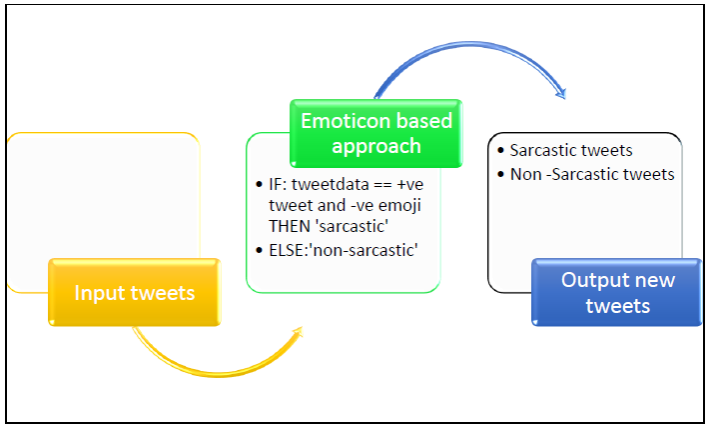

Figure 6: Emoticon based approach

4.6. Rule 6: Hierarchical Approach

Hybrid Based:

The sixth rule in the rule based approach is hierarchical approach. It's a sequential logic of all the above approaches (refer to figure 6). The tweets are marked as sarcastic if it's found true in anyone of the rule except for in hashtag rule (Since #tag rule tags all tweets as sarcastic) the implementation is simplified using hierarchical directory format and shown in the figure 7

Input: Processed 'Tweets'

Logic:

For *every-ruletweet* in *Allrules:*
Import *tweets*
From *tweets* import *tweet*
For each *tweet* in *tweets:*
For *data* in *tweets:*
#for emoticon, hyperbole,n-gram and contrast approach
If *everyrule (data) == 'sarcastic'*:
Append *'Sarcastic'* to [*Output-list*]:
Else:
Append *'Non-Sarcastic'* to [*Output-list*]:
For *tweet* in *tweets*:
For *listdata* in *list:*
Newlist Join (*tweet* with *Output-List)*
Output*: [Newlist]* of tweets

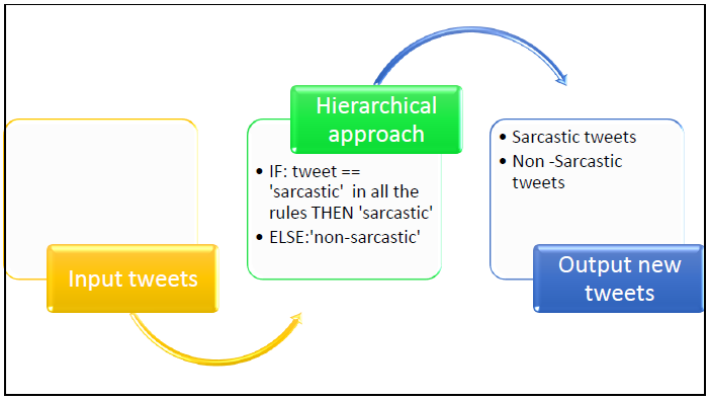

Figure 7: Hierarchical Approach

Hierarchical Plotting table: If the tweets are marked as sarcastic for a sequential logic of contrast with lexicon, hyperbole and emoticon approaches. Then tweet is automatically considered as sarcastic and tagged under hierarchical rule as shown in fig 8.

Tweets	Contrast	Lexicon	Emoticon	Hyperbole	Hierarchical
Tweet1	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic	Sarcastic	Sarcastic
Tweet2	Non-Sarcastic	Non-Sarcastic	Sarcastic	Non-Sarcastic	Sarcastic
Tweet3	Non-Sarcastic	Sarcastic	Non-Sarcastic	Non-Sarcastic	Sarcastic

Tweet4	Sarcastic	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic	Sarcastic
Tweet5	Non-Sarcastic	Non-Sarcastic	Sarcastic	Sarcastic	Sarcastic
Tweet6	Sarcastic	Sarcastic	Non-Sarcastic	Non-Sarcastic	Sarcastic
Tweet7	Sarcastic	Sarcastic	Sarcastic	Non-Sarcastic	Sarcastic
Tweet8	Non-Sarcastic	Sarcastic	Sarcastic	Sarcastic	Sarcastic
Tweet9	Sarcastic	Non-Sarcastic	Sarcastic	Sarcastic	Sarcastic
Tweet10	Non-Sarcastic	Sarcastic	Sarcastic	Sarcastic	Sarcastic
Tweet11	Sarcastic	Sarcastic	Non-Sarcastic	Sarcastic	Sarcastic
Tweet12	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic

Figure 8: Hierarchical plotting table

4.7. Rule 7: Hybrid Approach

Hybrid Based:

The final rule in the rule based approach is hybrid approach. It's a combinational logic of all the above approaches (refer to figure 7). The tweets are marked as sarcastic if it's found true in anyone of the rule except for in hashtag rule (Since #tag rule tags all tweets as sarcastic) the implementation is simplified into hybrid directory format and shown in the figure 9

Input: Processed 'Tweets'

Logic:

For *every-ruletweet* in *Allrules:*
  Import *tweets*
  From *tweets* import *tweet*
For each *tweet* in *tweets:*
    For *data* in *tweets:*
#for emoticon, hyperbole,n-gram and contrast approach
 If *tworules (data) == 'sarcastic'*:
 Append *'Sarcastic'* to [*Output-list*]:
  Else:
    Append *'Non-Sarcastic'* to [*Output-list*]:
For *tweet* in *tweets*:
  For *listdata* in *list:*
Newlist Join (*tweet* with *Output-List)*
 Output*: [Newlist]* of tweets

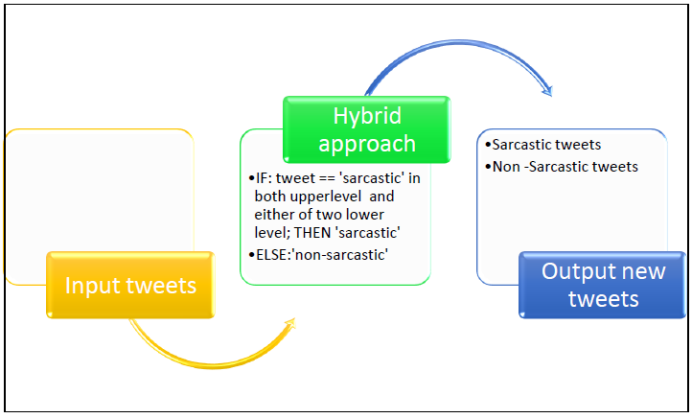

Figure 9: Hybrid Approach

Hybrid Plotting table: Tweets are marked as sarcastic for a combinational logic for either contrast with hyperbole, or contrast with lexicon, or lexicon with hyperbole, or lexicon with emoticon or contrast with emoticon and emoticon with hyperbole then tweet is automatically considered as sarcastic and tagged under hybrid rule as shown in fig 10.

Tweets	Contrast	Lexicon	Emoticon	Hyperbole	Hybrid
Tweet1	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic	Sarcastic	Non-Sarcastic
Tweet2	Non-Sarcastic	Non-Sarcastic	Sarcastic	Non-Sarcastic	Non-Sarcastic
Tweet3	Non-Sarcastic	Sarcastic	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic
Tweet4	Sarcastic	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic

Tweet5	Non-Sarcastic	Non-Sarcastic	Sarcastic	Sarcastic	Sarcastic
Tweet6	Sarcastic	Sarcastic	Non-Sarcastic	Non-Sarcastic	Sarcastic
Tweet7	Sarcastic	Sarcastic	Sarcastic	Non-Sarcastic	Sarcastic
Tweet8	Non-Sarcastic	Sarcastic	Sarcastic	Sarcastic	Sarcastic
Tweet9	Sarcastic	Non-Sarcastic	Sarcastic	Sarcastic	Sarcastic
Tweet10	Non-Sarcastic	Sarcastic	Sarcastic	Sarcastic	Sarcastic
Tweet11	Sarcastic	Sarcastic	Non-Sarcastic	Sarcastic	Sarcastic
Tweet12	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic	Non-Sarcastic

Figure 10: Hybrid plotting table

V. RESULTS AND DISCUSSION

The output of rule based approach is a list of tweets marked either sarcastic or non-sarcastic for every rule in a rule engine and each result were saved by their rules name respectively as ".csv" files and charts were produced automatically by python and plotly [3]. In figure 8 a comparative analysis has been shown for the Rule based approach. And proving that hybrid approach yields a better result compare to other approaches. The Rule based approach was able to detect sarcasm in text over 80% at precision rate which proved that it was giving 3% more accurate results compare to the existing algorithm [10] in few test runs. Test runs for all the features in rule engine is represented through charts as shown in fig 11, fig 12, fig 13, fig 14, fig 15, fig 16, fig 17 and fig 18 Fig. 3.6.1 displays a comparative analysis chart of all the rules in our rule based approach. Fig. 3.6.2 states about contrast approach this is one of the upper layer rule which is based on features such as POS tags and found that over 40% of the time tweets written comes under this category. Fig. 13 tells about the emoticon rules as observed not everyone uses emoticons all the time and the results had a very less complexity and time while parsing this phase but it did play a vital role in hybrid and hierarchical approach. Fig. 3.6.4 shows the results of hash tag rule which is considered to be by default rule results contains tweets as sarcastic and non-sarcastic if there exists "#sarcasm". Fig. 3.6.5 and Fig.3.6.6 shows results of lexicon and hyperbole approaches respectively. Fig. 3.6.7 and Fig. 3.6.8 shows the results of combinational logic and sequential logic of hybrid and hierarchical approach. From all the results after working over three different sets of sarcastic set of tweets we come to a conclusion that the contrast approach, lexicon, hyperbole and hybrid approaches are considered to be the higher level approach to give accurate and precise results hence deducing a better prediction rule compare to previous work.

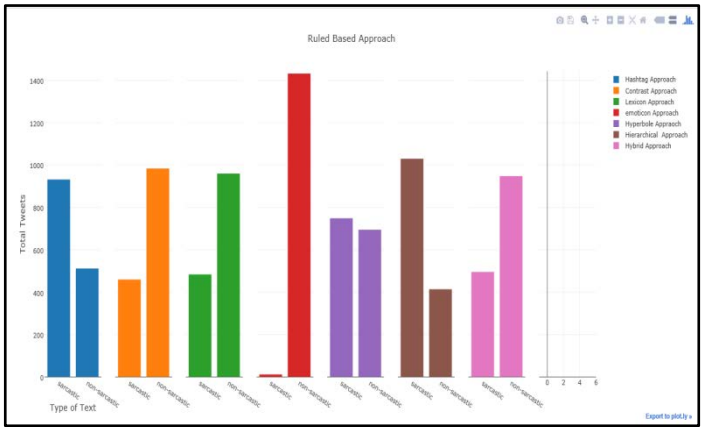

Figure 11: (comparative analysis chart) Rule based approach

Charts of the 7 rules in Rule Based Approach:

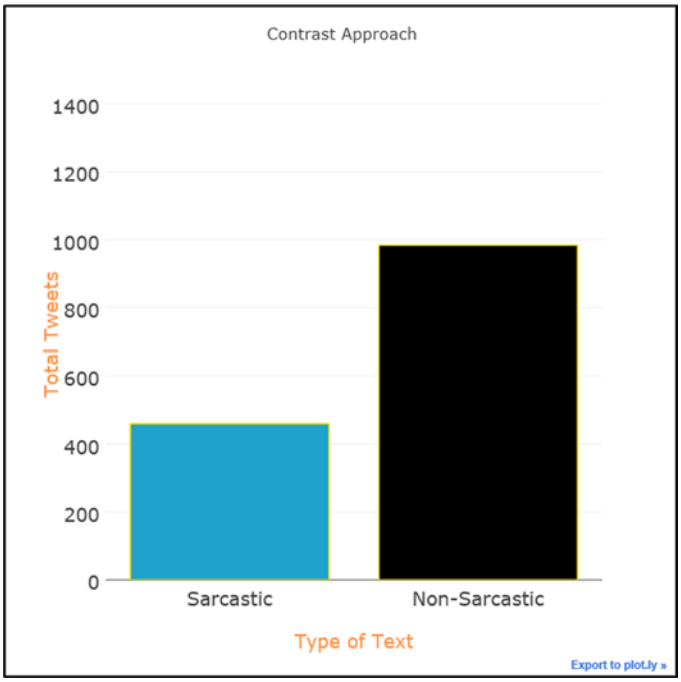

Figure 12: Contrast Approach

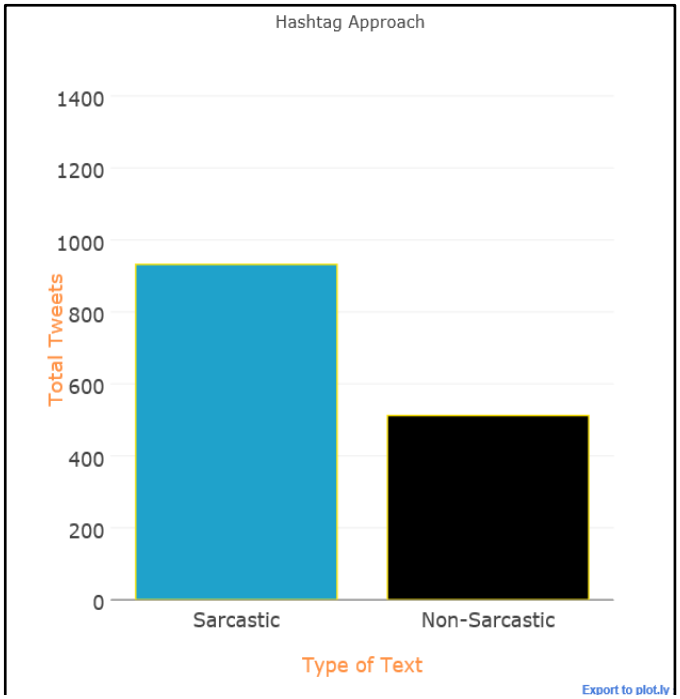

Figure 14: Hashtag Approach

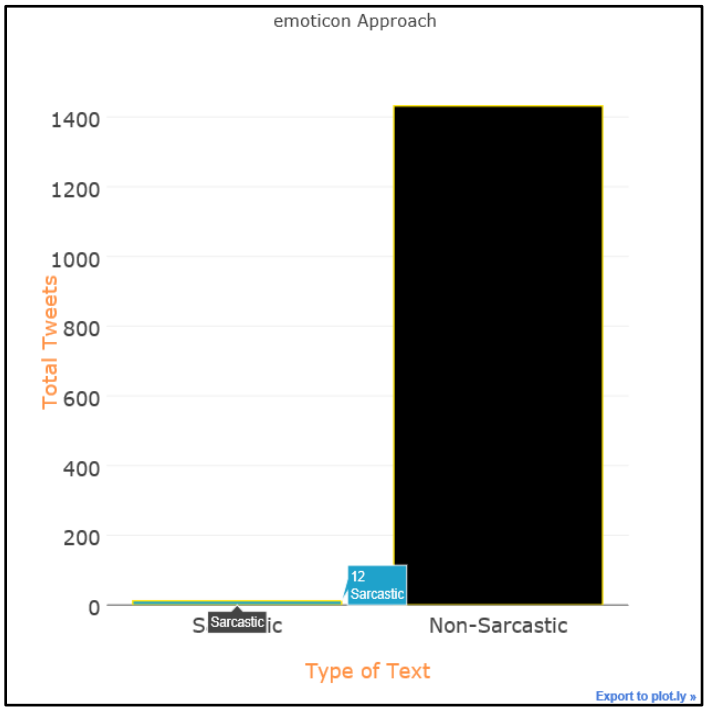

Figure 13: Emoticon Approach

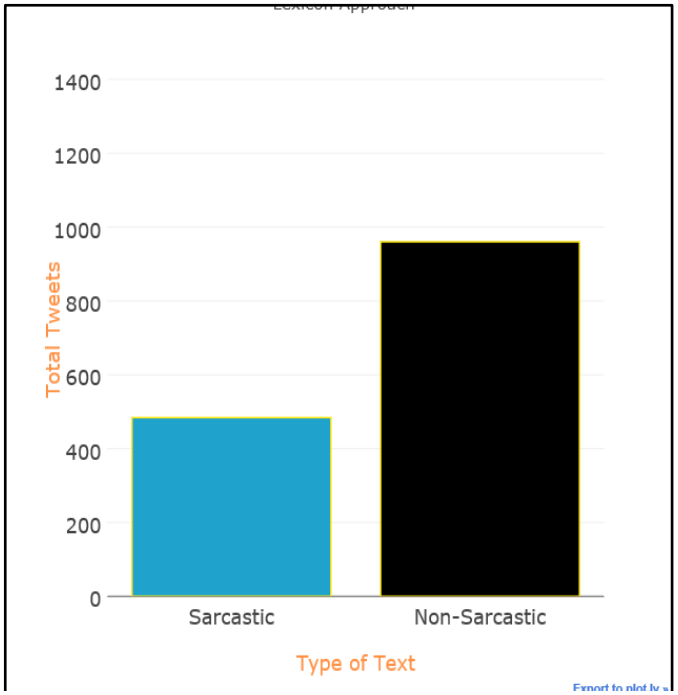

Figure 15: Lexicon Approach

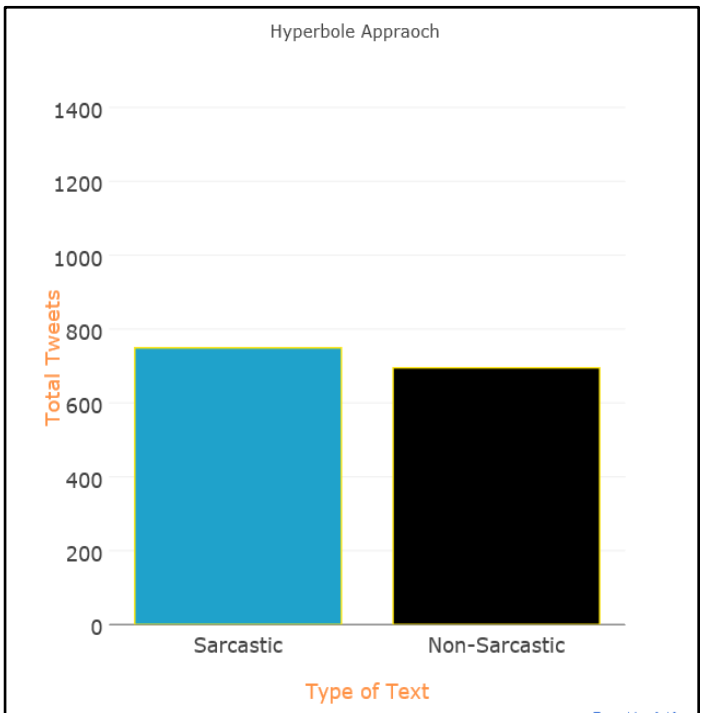

Figure 16: Hyperbole Approach

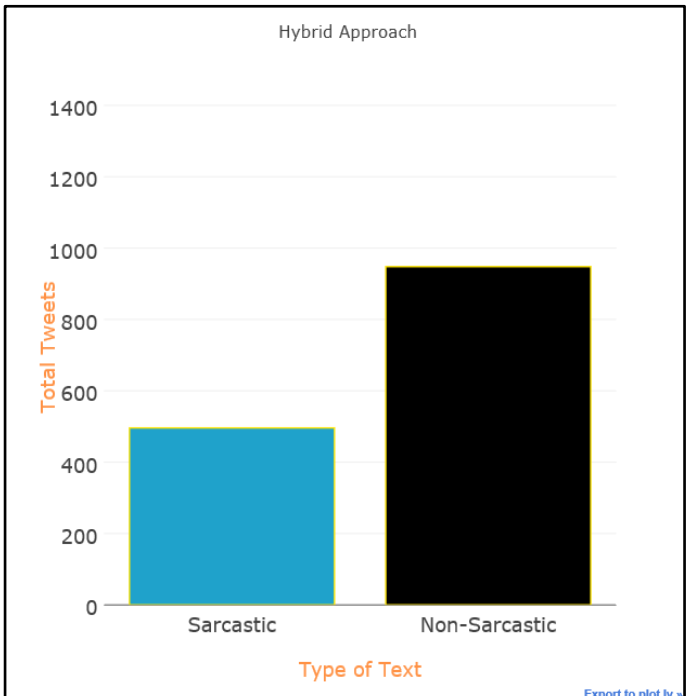

Figure 17: Hybrid Approach

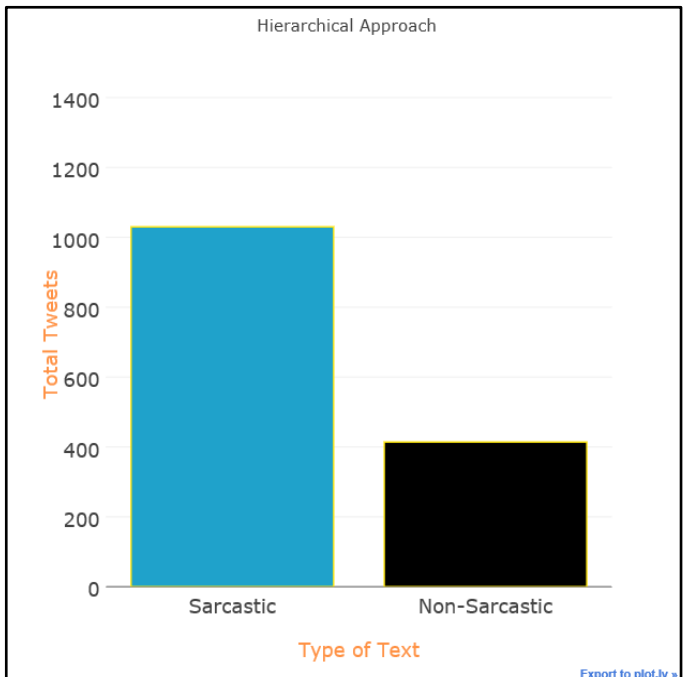

Figure 18: Hierarchical Approach

VI. CONCLUSION

Sarcasm detection on twitter tweets is more complicated as it provides very less detailed results, and developing a dictionary for these kind of text documents takes more time and resources. Social media posts are hard to analyze on the phrase or sentence level because of theirs unique structure and grammar. Since twitter allows user to enter 140 characters processing time also increases. The sarcasm detection was ignored for different languages (except English), repeated tweets and empty or a single letter/word tweets. The future work will be focused on backtracking of tweets (analyzed based on user's past replies and comments) and multilingual language support.

REFERENCES

[1] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis, University of Pittsburgh, Pennsylvania, 2005

[2] J. Zhao, L. Dong, J. Wu, and K. Xu, "Moodlens: An emoticon-based sentiment analysis system for Chinese tweets," in Proceedings of the18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2012, pp. 1528–1531.

[3] Plotly.blogspot.org

[4] Ms. HarvinderJeetKaur, Mr. Rajiv Kumar," "Sentiment Analysis from Social Media in Crisis Situations" ICCCA, 2015

[5] Barbosa, L., and Feng, J. "Robust sentiment detection on twitter from biased and noisy data." In Proceedings of COLING, pp. 36–44, 2010.

[6] Ms. HarvinderJeetKaur, Mr. Rajiv Kumar, "Sentiment Analysis from Social Media in Crisis Situations" ICCCA, 2015

[7] O. Medelyan, S. Schulz, J. Paetzold, M. Poprat, and K. Marko. "Language specific and topic focused web crawling." In Proceedings of the Language Resources Conference LREC. 2006.

[8] N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, and A. Iamnitchi, "Prometheus: user-controlled p2p social data management for socially-aware applications," in Proceedings of the ACM/IFIP/USENIX 11th International Conference onMiddleware, ser. Middleware '10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 212 231. [Online]. Available: http://dl.acm.org/citation.cfm?id=2023718.2023733

[9] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonalves, S. Patil, A. Flammini, and F. Menczer, "Truthy: mapping the spread of astroturf in microblog streams," in Proceedings of the 20thInternational Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1, 2011 (CompanionVolume), S. Srinivasan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, and R. Kumar, Eds. ACM, 2011, pp. 249–252.

[10]B. Pang and L. Lee, "Opinion mining and sentiment analysis," Found. Trends Inf. Retr., vol. 2, no. 1-2, pp. 1–135, Jan. 2008

[11] M. Hu and B. Liu, "Mining and summarizing customer reviews," in Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: ACM, 2004, pp. 168–177.

[12] Y. Rao, J. Lei, L. Wenyin, Q. Li, and M. Chen, "Building emotional dictionary for sentiment analysis of online news," World Wide Web, vol. 17, pp. 723-742, Jun. 2014.

[13] Y. Bae and H. Lee, "Sentiment analysis of Twitter audiences: Measuring the positive or negative influence of popular twitterers," Journal of the American Society for Information Science and Technology, vol. 63, no. 12, pp. 2521-2535, 2012.

[14] P. Gonçalves and M. Araújo, "Comparing and combining sentimentanalysis methods," In Proceedings of the first ACM conference on Online social networks. ACM., pp. 27-38, 2013.

[15] Go, A., Bhyani, R., and Huang, L. "Twitter sentiment classification using distant supervision." CS224N Project Report, Stanford, 2009.

[16] N. Jakob and I. Gurevych, "Extracting opinion targets in a single- and cross-domain setting with conditional random fields," in Proceedingsof the 2010 Conference on Empirical Methods in Natural LanguageProcessing. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 1035–1045.

[17] HarunaIsah, Paul Trundle, Daniel Neagu "Social Media Analysis for Product Safety using Text Mining and Sentiment Analysis"