International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 04 | Apr-2018 www.irjet.net p-ISSN: 2395-0072

A SYSTEM FOR DETERMINING SARCASM IN TWEETS: SARCASM DETECTOR

Shweta Gupta[1], Malav Shah[2], Sanket Shah[3], Ashwini Deshmukh[4]

[1,2,3] Department of Information Technology, Shah and Anchor Kutchhi Engineering College, Mumbai, India.
[4]Assistant Professor Department of Information Technology, Shah and Anchor Kutchhi Engineering College, Mumbai, India.

Abstract - *Sarcasm is form of irony which conveys an opinion or a sentiment about a particular target object. Feedback of the customers are more essential for a particular product to know about the flaws and the criteria in which they need to improve. It is basically identified by a user's tone of speech and certain gestures which indirectly expresses sarcasm.*

Since the conventional approaches of collecting feedbacks through interviews and questionnaires was too time consuming, so we propose an approach of a desktop application which would help in analyzing the product reviews and further help in improving the products. A dictionary of words will be created in the basis of sentiment analysis and polarity of a statement. Based on the dictionary the tweet is classified as sarcastic or non-sarcastic.

Key Words*: *PBGLA, POS Tagging, Parse Tree, Sentiment Analysis.

1.INTRODUCTION

Twitter has become one of the biggest web destinations for users to express their opinions and thoughts. Many companies and organizations have been interested in these data for the purpose of studying the opinion of people towards political events, popular products or movies. Throughout the last few years, Twitter content continued to increase: hundreds of millions of tweets are sent everyday by more than 285 million active users.

Furthermore, presence of sarcasm makes the task even more challenging: sarcasm is when a person says something different from what he means. However, due to the informal language used in Twitter and the limitation in terms of characters (i.e., 140 characters per tweet), understanding the opinions of users and performing such analysis is quite difficult. Some people are more sarcastic than others, in general, sarcasm is very common, though, difficult to recognize.

2.SARCASM DETECTION

Sarcasm can be manifested in many different ways, but recognizing sarcasm is important for natural language processing to avoid misinterpreting sarcastic statements as literal. Sarcasm is generally characterized as ironic wit that is moreover intended to insult, amuse or mock. For example, sentiment analysis can be easily misled by the presence of words that have a strong polarity but are used sarcastically, which means that the opposite polarity was intended. Consider the following tweet on Twitter, which includes the words "yay" and "thrilled" but actually expresses a negative sentiment: "yay! it's a holiday weekend and I'm on call for work! couldn't be more thrilled!"- *Sarcastic statement.* In this case, it reveals the intended sarcasm, but we don't always have the benefit of an explicit sarcasm label.

(a) Yes, let's all cheer on the glorified shameless point-farming bitch of this site.

(b) I love how you are using violence to threaten people. when, your argument is disputing violence.

(c) I love how scum barge doesn't give a rebuttal to this. he knows you have him by the balls.

The sarcasm in these tweets arises from the juxtaposition of a positive sentiment word (e.g., love, cheer) with a negative activity or state (e.g. Rebuttal, shameless, violence).

2.1 Part-of-Speech (POS) Tagging

It is a python-based package working on NLTK (Natural Language Toolkit), and freely available on the internet. Penn Treebank Tag Set notations such as JJ-adjective, NN-noun, RB-adverb, VB-verb, and UH-interjection, etc. were used. It is a process of taking a word from text (corpus) as input and assign corresponding part-of-speech to each word as output based on its definition and context *i.e.* relationship with adjacent and related words in a phrase, sentence, or paragraph. In this paper, TEXTBLOB was deployed to calculate POS tagging and parsing of the given text.

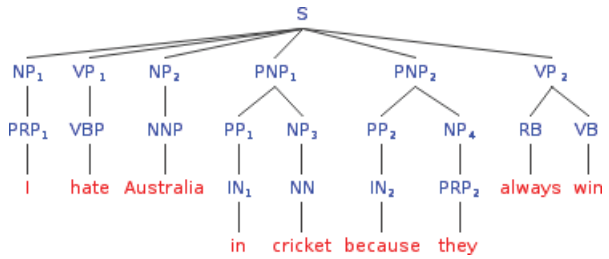
2.2 Parsing and parse tree

The term parse tree itself in computational linguistics is primarily used in theoretical syntax, the term syntax tree is more common. A parse tree or parsing tree or derivation tree or concrete syntax tree is an ordered, rooted tree that represents the syntactic structure of a string according to some context-free grammar.

Parse trees are usually constructed based on either the constituency relation of constituency grammars (phrase structure grammars) or the dependency relation of dependency grammars. Parse trees may be generated for sentences in natural languages (see natural language processing), as well as during processing of computer languages, such as programming languages.

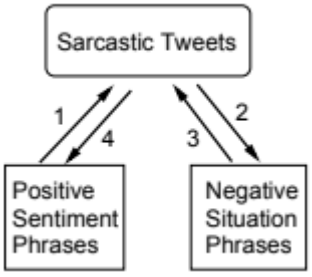
To generate a parse tree, one requires the pairing data. In our project to find parsed data of tweets, TEXTBLOB was used. Based on this sentence the parsed tree is shown below

"I hate Australia in cricket because they always win"



3. PROPOSED SCHEME

This section deals with goals and proposed algorithms. To identify sarcasm, we observed that the sentence is sarcastic when there are words showing contradictory sentiments in desired situations.



This algorithm which is designed to detect the sarcasm detection in tweets. Where we use parsing based lexical

generation algorithm(PBLGA) to find out the lexicons in four categories, namely positive sentiment, negative sentiment, positive situation and negative situation. After this on the basis of the tweets sarcasm would be found by (a) contradiction of negative sentiment and positive situation (b) contradiction of positive sentiment and negative situation. The algorithm is given below.

PBGLA:

Input: Tweet Corpus(C)
Initialization Output: Sarcastic or non-sarcastic

Notations: T=Tweet, C= Corpus, P = Phrase , TWP=Tweet wise parse, TF: Tweet File,

PF=Parse file, Adj= Adjective phrase , Adv=Adverb phrase, Noun: Noun phrase, Verb=Verb phrase, sent_pos = Positive Sentiment, sent_neg = Negative Sentiment, sit_pos = Positive Situation, sit_neg = Negative Situation, SC: Sentiment Score, Sentiment= Sentiment File, Situation= Situation File.

: Sentiment = { Ø }, Situation = { Ø }, sent_pos = { Ø }, sent_neg = { Ø }, sit_pos = { Ø}, sit_neg = { Ø }.

Algorithm:

for T in C **do**
k = find_parse (T)
PF ← TF ∪ k
end for
for TWP in PF **do**
k = find_subset (TWP)
if k = Noun || Adj || (Noun + Verb) **then**
Sentiment ← Sentiment ∪ k
else k = Verb || (Adv + Verb) || (Verb + Adv) || (Adj + Verb) || (Verb + Noun) || (Verb + Adv +Adj) || (Verb +Adj +Noun) || (ADVP +Adj + Noun) **then**
Situation ← Situation ∪ k
end if
end for
for *P* in Sentiment **do**
SC = sentiment_score (*P*)
if SC >0.0 then
sent_pos ←sit_pos ∪ *P*
else SC <0.0 then
sent_neg← sit_neg ∪ *P*
end if

end for
for P in Situation **do**
SC = sentiment_score (P)
if SC >0.0 then
sit_pos ← sit_pos ∪ P
else SC <0.0 then
sit_neg ← sit_neg ∪ P
end if
end for

```
for (tweets in dataset) do
 count=0
 SarcasmFlag=False
 for (words in tweets) do
  if ((word==any phrase in sent_pos)&&(count==0))
 count=1
  Check (sit_neg)
    continue
  end if
  If((word==any phrase in sit_neg)&&(count==1))
    SarcasmFlag = True
    Break
  end if
  Else if((word==any phrase in sent_neg)&&(count==0))
       count=1
       Check (sit_pos)
       Continue
       end if
      If((word==any phrase in sit_pos)&&(count==1))
        SarcasmFlag = True
        break
        end if
 end for
end for
```

A parse tree divides a text into a set of noun phrase (NP), verb phrase (VP), Adjective phrase (ADJP), and adverb phrase (ADVP, etc. According to Penn Treebank, verb (V) can be in any form as VB, VBD, VBG, VBN, VBP, and VBZ. Adverb (ADV) can be in any form as RB, RBR, WRB, and RBS. Adjective can take any form as JJ, JJR, and JJS. Noun (N) can take any form as NN, NNS, NNP, and NNPS.

In the proposed algorithm, the corpus of tweets are passed as an input, which is saved in the parsed file. If The subset of the parsed file contains noun phrase followed by verb phrase or adjective phrase, it is added in the sentiment file. Similarly, the verb phrase or (adverb phrase followed by verb phrase) or (verb phrase followed by adverb phrase) or (adjective phrase followed by verb phrase) or (verb phrase followed by noun phrase) or (verb phrase followed by adjective phrase followed by noun phrase) or (adverb phrase followed by adjective phrase followed by noun phrase) or (verb phrase followed by adverb phrase followed by adjective phrase) adds to the situation file.

The sentiment score gives us the numerical representation which is more precise of the sentiment polarity. See below to understand the dictionary and how it is divided into different categories. Sentiment scores and polarity are available for documents, named entities, themes, and queries.

SC (Sentiment Score) = PR – NR
Where SC = Sentiment Score
PR = Positive words / Total number of words
NR = Negative words / Total number of words

On the basis of the sentiment score the dictionary is produced based on which we have extended the algorithm which analysis the input and would give the desired output

4.RESULTS AND ANALYSIS

After the algorithm is implemented based on the given the conditions the dictionary which is produced will be in four categories which is as shown in the table below.

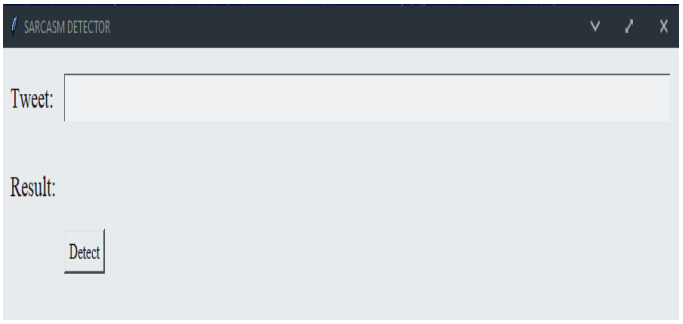
Positive sentiment phrase: Accept, approve, cute, luck, nice, happy, glad, support, best, Awesome, Perfect, joy, strong, Pretty, proud, hilarious, Better, gorgeous, honest, innocent, talented, excellent, lovely, outstanding, bright, elegant, easy, supportive, attractive, huge, interesting, successful, healthy, famous, pleasant, beloved, enjoy, appreciate, holy, support etc.
Negative sentiment phrase: Reject, criminal, least, lost, terrible, little bit, ugly, excuse, dirty, little, expensive, half, cold food, tight jeans, troubled, least, hard, rude, wrong, dramatic, abusive, unhealthy, crap, bad, a few days, sick, dumb, pathetic, victim, useless, fluffy, violent, Poor, brutal, worst, crazy etc.
Positive situation phrase: regret, love, love seeing, just love discovering, just love, effectively making, just love living absolutely LOVE, feel so loved, wanna be loved, winning, love falling, honestly tell, will fly, now enjoying managing, most entertaining, Enjoying, thrilled, Delighted, really enjoying, really is thrilled, are winning, most enjoy, most loved, enjoyed looking, feel so satisfied, Proudly displayed, have liked, so impressed, wisely locked, is absolutely delighted, Just loved getting being brilliantly led, feel loved, greatly appreciate, gladly appreciate, most appreciate, sincerely appreciate, very excited, good news etc.
Negative situation phrase: Clarifying, are pumped, will be re- leased, are arriving, babysitting, only run, kicking, gets stuck, is losing, crashing, destroyed, attacking, criticizing, is lying, biting, confused, dividing, exhausted, keep arguing, shouting, awkwardly standing, are limping, needs help, forgotten, can barely walk, already upset, crying, get dropped, feel tired, banning, stole, so upset etc.

As proposed by our algorithm that if there is a contradiction between positive sentiment and negative situation or negative sentiment or a positive situation it would analyse the statement and find the words which are obtained in the dictionary. If the words are found which

matches the condition, it will give an output as the sentence sarcastic or else the sentence is non- sarcastic.

5. IMPLEMENTATION

The working of Sarcasm Detection is quite simple. The user has to give a tweet after which it will analyze and detect the
sarcasm in the given tweet. The GUI will be displayed as shown below:



Now, the user can enter a tweet in the Textbox below. Here the sentence is " I love how you are using violence to threaten people. When, your argument is disputing violence." The words love (Positive Sentiment) and threaten (negative situation) contradicts the statement, so the output is sarcastic.

Here the input given is "You blame others for your failure to understand the basics." This tweet is a normal sentence so it is detected as non-sarcastic statement.

6. CONCLUSION

In this analysis, we considered various reviews which were given about different products which will help in extracting the concerned tweets of the respective products.

The lexicon approach provided with the polarity of the words with positive or negative words are distinguished accordingly. The Natural Language Toolkit(NLTK) deals with the various problems faced in differentiating between the words, i.e., noun or adverbs, adjectives which helped us to improve total accuracy of the results obtained. A number of works has been done for informal reviews and blogs. But we look forward to product services and help those services to obtain genuine reviews on their products. The previous stages comprised of developing the system features a working on the core product which will fulfill all the essential features. This application interface to gain results for the application developers with the ability to make it easier for the product companies to utilize this feature of the system.

7.ACKNOWLEDGEMENTS

We sincerely express our deepest gratitude to our Principal Dr. Bhavesh Patel, our Head of Department Prof. Swati Deshpande and project supervisor Prof. Ashwini Deshmukh. We were fortunate to have met such supervisors. We like to express our thankfulness for their kind supervision and offering unfailing support, invaluable advices and comments and helpful and useful discussions in selecting the interesting point and during the preparation of this BIG project. We owe A special acknowledgment to them for giving us a lot of their time during the period of preparing this project. We could never had done it without their support, technical advice and suggestions, thorough reading of all our work.

7. REFERENCES

[1] Tomoaki_Ohtsuki --Sarcasm Detection in Twitter "All your products are incredibly amazing!!!" – Are They Really? IEEE, December 2014.

[2] Ellen Riloff --Sarcasm as Contrast between a Positive Sentiment and Negative Situation. ACL,pp. 704-714,2013.

[3] Ashwin Rajadesingan --Sarcasm Detection on Twitter: A Behavioral Modeling Approach. ACM, pp. 97-106, February 2015.

[4] Santosh Kumar Bharti --Parsing-based Sarcasm Sentiment Recognition in Twitter Data. IEEE/ACM, pp. 1373-1380, August 2015.

[5] Satoshi Hiai --A Sarcasm Extraction Method Based on Patterns of Evaluation Expressions. IEEE, pp. 31 - 36, July 2016.