Pragmatic Feature-Based Approach for Sarcasm Detection using Hadoop

Galigutta Chaithanya*[1], A. Nagaraja Rao[2]

[1]School of Computer Science and EngineeringVellore Institute of TechnologyVellore, India
[2]School of Computer Science and EngineeringVellore Institute of TechnologyVellore, India
[1]galigutta.chaithanya2019@vitstudent.ac.in

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

Abstract*:* People often express their thoughts which are evil by utilising superb or intensified excessive exceptional phrases in the text, this type of sentiment is called Sarcasm. While talking, all human beings regularly use intense stress while talking and some gestural clues like moving eyes, movement of hands, etc., to symbolise Sarcasm. These physical gestures are missing in the textual data and making sarcasm dectection very difficult. Due to these difficulties, researchers are showing high interest in detecting Sarcasm in text, mainly in emotion and textual content material. In this project, we detect Sarcasm from both TEXT and EMOTICONS by using the Hadoop MapReduce framework. Sarcasm means one person commenting on other people by using both positive and negative words

Keywords: sentiment, Hadoop MapReduce, Emoticons

1. Introduction

In the existing world, hundreds of plenty of human beings are spending valuable time on OSN to share their data, talk about and be a section of with every other. OSN is accountable for producing information at an unparalleled rate. For a few years, the author's have been working on extraordinary hassle extracted from OSN mining, like protection challenge with user's non-public data, sample recognition, sentiment distinction etc. Sentiment Analysis is an effective methodology, has its own drawbacks, sarcasm is one among them. Sarcasm is a kind of sentiment the region human beings precise their ideas verbally or nonverbally with a modified polarity which functionality they painting their feelings which are opposite to their relevant feeling—something which is contrary to their supposed meaning. Hence, computerised Sarcasm detection from OSN turns into a challenge that has computational techniques that as rules to predict if a given textual content material fabric is sarcastic or non-sarcastic. Merriam-Webster1 definition for Sarcasm as the use of phrases that endorse the contrary of what people barring a doubt, pick to say regularly to insult someone, to exhibit irritation, or to be witty. So, to take a seem at Sarcasm desirable, a laptop computer PC would have to mother or father defamed and hazardous remarks for gorgeous users. Sarcasm can be used for stunning purposes, like humiliation or criticism. Firstly, computing laptop computer takes a piece of textual content material cloth material fabric as entering, and the output is to predict whether or not or no longer or no longer or now now now not or now now now no longer it is sarcastic.

2. Literature Survey

Here Literature Survey is done in two stages. At first, Data Processing from different sources, which include real-time twitter data capturing and pre-processing, surveyed and from data libraries. Sarcasm detection is done thereafter

A. Large volume capturing and pre-processing of tweets

Rapid growth in the social networking platforms and adaptation of data encourage consumers to produce data at very rapid rate. Huge data sets Processing and storing become a critical problem. The main social networking platforms is Twitter which continuously produces data. In the previous literature, as we can observe, a huge number of the researchers and scholars have been using Tweepy and Twitter4J for consolidation of Twitter tweets [1]. The Twitter Application Interface (API) enable the feature of streaming API to obtain real-time data access to Twitter tweets. People like [11] Befit and Frank discuss the problems and limitations in data stream capturing from Twitter. Twitter has some limitations on the data usage and data retrieval from their Twitter retrieval API, and anyone can download only a feasible quantity of tweets at a fixed time frame with the help their libraries and APIs. Researchers are working on the Hadoop ecosystem [6] for processing and storing huge amounts of Twitter data from Twitter. There is a necessity for promising techniques to get a huge amount of Twitter data from Twitter. Tufekci and Zeynep [12] examined the theoratical methodologis and problems related to social media data operations for the big data analysis. Shirahatti et al. [13], in their research, used the Hadoop ecosystem along with Apache Flume to gather and collect twitter data from Twitter. Taylor et al. [15], in their applications, used the Hadoop framework in the bioinformatics domain. Any system where large data should be processed used the Hadoop framework and related tools and engines to work on the data captured and processed accordingly

B. Classification based on pragmatic features

Its been frequent in tweets and messages usage of the figurative text and symbols due to the length of message constraint. These symbols and figurative text data which consist of smilies, emoticons, replies, etc are called pragmatic features. Several authors and researchers have used this attribute in their work to identify Sarcasm as it is the main attributes. Carvalho et al. [7] pragmatic features from the newspapers like special punctuations and context related to emotions are used to detect contradiction. Pragmatic features, including figurative text, is one of the main attribute to detect Sarcasm in text. Furthur research has been done on this features with repilies having smilies and emoticons and building a system which detects the sarcasm with the help of pragamatic feature based analysis. Tayal et al. [17] also used this pragmatic feature in Twitter data related to politics to predict which party will be a win in the election. Similarly, Rajade- singan et al. [18] used behavioural and psychological features on the user's past and present tweets to detect Sarcasm.

C. **Sarcasm detection**

Justo et al. (2014) have developed a laptop computer pc laptop that robotically classifies nasty and sarcastic utterances the utilisation of supervised getting to apprehend techniques. For sarcastic utterance creator introduced furnish up provide up quit stop end result with more accuracy and semantic records sources and pattern of files is sufficient to be aware of nasty language. Generally, it opinions that every and each emotion has one of a range of exceptional factors that will provide a truly useful beneficial useful resource with the computing gadget to apprehend them

Kunneman et al. (2014) laboured on a bunch of hashtag tweets denoting Sarcasm and gave grant up end cease end result that explicitly marked hashtags reduces the in related hashtags warning symptoms and signs and symptoms and signs and symptoms a polarity alternate in most of the cases. Sarcasm detection being the optimum goal and finding usages in the real time data whether its from the tweeter or any other sources is the main objective for many scholars and reasearchers.

In their work, they detect Sarcasm from tweets utilising real-time stamping Bharti et al. (2016). Authors are conscious of the large difference in Sarcasm with the truly beneficial recommended resource of the utilisation of classifying tweets in reality, especially in particular primarily based absolutely except a doubt on three brilliant factors lexical features, hyperbole factors and pragmatic features. Hadoop in special frequently particularly pretty based totally absolutely framework used to be as rapidly as shortly as used and processed the utilisation of the MapReduce model to detect sarcasm and then techniques like POS tagging, parsing, textual content material fabric material mining, sentiment big difference are used to develop to be aware of Sarcasm with six proposed algorithm, especially PBLGA algorithm, IWS, PSWAP, TCUF, TCTDF, and LDC.

D. **Classification based on Hyperbole features**

Hyperbole feature another special category that is a challenge for sentiment analysis. It has characters and special text, which includes interjections, punctuations and intervening text in the sentence. The previous author used this methodology of hyperbole features and obtained good Precision and accuracy to in Twitter data. Research [19] elaborated punctuations and interjections and other hyperbolic features which explained how hyperbole will be the key aspect of sarcasm detection. Utsumi and Akira [41] explored complex adverbs and adjectives and how these two extremes intensifies the textual data. Quite often, it suggests an implicit way to view negative attitudes, i.e., Sarcasm. Filatova and Elena [11] discussed the hyperbole features in document level text data. They have concluded that the sentence or phase level will not sufficient for promising accuracy which leads to considering the text level context to improve accuracy.

3. Proposed algorithm

E. **Sarcasm analysis with *mapreduce* functions**

Here, the mapping methods consist of three approaches to detect Sarcasm. Intial interjection, hyperbole features and pragmatic feature analysis, yet, all these have similar initial process with a different set of rules applied.

1) Real-time tweets capturing procedure and pre-processing

API of The Twitter Streaming returns data in the form of JSON format to store in HDFS. Very complicated integration is required to exclude errors related to writing code and security. We prefer Cloudera Hadoop for usage. This enables us to directly store and retrieve into the HDFS. Usually, HDFS uses Apache flume to store

data. It is a data integration system that is sets tunnels by which which information channels between sources and sinks and other components that are linked. Flume considers each chunk of data as an event which passes through the tunnel. The Twitter API is now the source, and the server being the sink which writes the information that has been passed out to the HDFS.

Once HDFS is filled with the data from Twitter API, data pre-processing can only be done after converting the JSON format data into usable text format. Oozie module is used to handle the work flow, that runs at periodic intervals. Oozie is configured based upon periodic recoveries usually hourly and store it in the hive after partitioning the data in the. The hive is a framework that enable one to convert the data and the data is loaded with the help of the Serializer–Deserializer which makes proccess easy to render the JSON formate data into a query-able format that helps us to work on the data. We then add back all these results as entries into the HDFS.

2) Lexicon generation algorithm based on parsing

Parsing based lexicon generation algorithm (PBLGA), which we have been known from earlier studies. From the HDFS stored tweets data are taken as inputs, which parses sentences as verb phrase (VP), noun phrase (NP), adjective phrase (ADJP) and more which are stored in the the phrase file for processing. Afterwards, phrase segregation is done using a rule-based classifier Fig. 1 furthur captures it into situtional phrase and sentiment phrase files

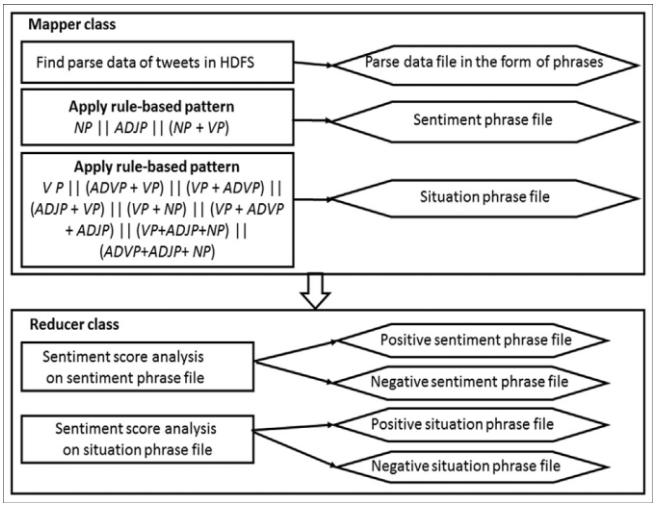

Fig. 1. Obtaining sentiment and situation phrases

The result from the mapper class after rule-based classification from the phrase file sent as an input to another class called reducer class. This class evaluates each phrase sentiment score in sentiment and as well as situational phrase file. The result for each phrase will be combined positive or negative score. Thus the phrase file is filled with phrases depending on the score, i.e. positive or negative, as mentioned in the Fig. 1. PBLGA produces a total of four files as an output, i.e.

Positive sentiment, positive situation, negative sentiment and negative situation. Afterwards we use those files to detect the structure contradiction in the data between negative situation and positive sentiment, negative sentiment and positive situation.

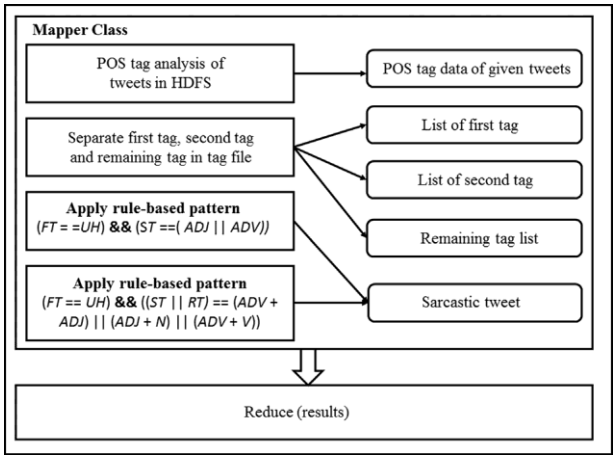

Fig. 2. Sarcasm detection procedure in tweets.

3) *Sarcasam detection based on EMOTICONS*

So far we have seen the sentiment analysis, sentence tokenising, POS tagging. Now we analyize the sarcasem based on the pragmatic feature called Emoticons. Here the entire process and algorithm remains same but we add preproccessed emoticons to interjections and process the while algorithm with different rule set as shown in Algorithm 2.

Algorithm 1. PBLGA_testing

Data: *dataset* := Tweets for testing, bags of lexicons
Result: *classification*:=sarcastic or non sarcastic

```
while tweets in dataset do
    count = 0
    sarcamsFlag = False;
    while words in tweet do
        if word==(any phrase in positive
          sentiment lexicons) && (count == 0)
          then
            count = 1;
            check negative situation lexicons
            continue;
        end
        if word == (any phrase in negative
          situation lexicons) && (count == 1)
          then
            sarcasmFlag = True
            break;
        end
        else
            if word == (any phrase in negative
              sentiment lexicons) && (count == 0)
              then
                count = 1
                check positive situation lexicons
                continue;
            end
            if word == (any phrase in negative
              situation lexicons) && (count == 1)
              then
                sarcasmFlag = True
                break;
            end
        end
        if sarcasmFlag==True then
            Given tweet is sarcastic
        end
        else
            Given tweet is not sarcastic
        end
    end
end
```

Mapper class will do the POS tagging considering emoticons as textual context. In the rule based pattern we use Universal Fact File(UFF)

Algorithm 2: Tweet contradict with EMOTICONS using universal_facts

Data: *dataset* := Corpus of universal facts.
Result: *Result*:= A ⟨*Key*, *Value*⟩ pair
Notation: S: Subject, V: verb, O: Object, T: tweets, C: corpus, PF: parse file, TWP: tweet wise_parse_phrase, UFF: Universal_fact_file.
Initialization : $PF = \{\phi\}$, $UFF = \{\phi\}$
while T *in* C **do**
  p = find_parsing (T) $PF \leftarrow PF \cup p$
end
while TWP *in* PF **do**
  S = find_subject (TWP)
  V = find_verb (TWP)
  O = find_object (TWP)
  Key ← (S + V)
  Value ← O
  $UFF \leftarrow \langle Key, Value \rangle$
end

RESULTS

Hadoop framework used for processing large data either from real time or from corpus. We have achieved for both real time and corpus with above mentioned algorithms.

Using the algorithms and modifying them to the need gives us a customised sarcasm detection model for a pragmatic feature called EMOTICONS. As mentioned above, Emoticons are converted to text, and then POS tagged and analysed for further usage. These emoticons are of different types and have different face values. Setting up the values and modifying the rules gives promising results.

Recall, *Precision* and *F-score* are statistical parameters that are used to evaluate and analyse proposed approaches. *Recall* indicates how much amount of extracted data is re- levant, *and Precision* indicates how much amount of relevant data is identified correctly. *F-score* uses both recall and Precision and evaluated as the harmonic mean of *Precision* and *recall*. The formula for three statistical parameters are given below:

$$Precision = \frac{T_p}{T_p + F_p}$$

$$Recall = \frac{T_p}{T_p + F_n}$$

$$F - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Where T_p is true positive, F_p is false positive and F_n is false negative.

In Fig 3 screen in first column we can see sentence and in second column we can see whether that sentence is sarcastic or non-sarcastic. In above screen in selected column we can see sentence as ‘richardebaker no it is too big Im quite happy with the Kindle’ where richardebaker saying kindle is too big in size (negative sentiment) but still he is quite happy with kindle (positive sentiment) so this sentence is combination of both positive and negative expression so it will be consider as ‘sarcastic’. In above screen I cannot display emoticon so I am displaying name of detected emoticon as ‘sob (means crying), heart_eyes (means smiling) etc. So in every

sentence in last word of sentence you can see name of emoticon. Now click on 'View Detection Graph' button to get below graph

Sentence	Detection Type
Reading my kindle Love it Lee childs is good read heart_eyes	Non Sarcastic
Ok first assesment of the kindle it fucking rocks heart_eyes	Non Sarcastic
kenburbary Youll love your Kindle Ive had mine for a few months and never looked back The new big one is huge No ne...	Non Sarcastic
mikefish Fair enough But i have the Kindle and I think its perfect heart_eyes	Non Sarcastic
richardebaker no it is too big Im quite happy with the Kindle heart_eyes	Sarcastic
Fuck this economy I hate aig and their non loan given asses sob	Non Sarcastic
Jquery is my new best friend heart_eyes	Non Sarcastic
Loves twitter heart_eyes	Non Sarcastic
how can you not love Obama he makes jokes about himself heart_eyes	Sarcastic
Check this video out President Obama at the White House Correspondents Dinner stuck_out_tongue_winking_eye	Sarcastic
Karoli I firmly believe that ObamaPelosi have ZERO desire to be civil Its a charade and a slogan but they want to d...	Non Sarcastic
House Correspondents dinner was last night whoopi barbara amp sherri went Obama got a standing ovation heart_eyes	Non Sarcastic
Watchin EspnJus seen this new Nike Commerical with a Puppet Lebronsht was hilariousLMAO heart_eyes	Non Sarcastic
dear nike stop with the flywire that shit is a waste of science and ugly love vincentxx sob	Non Sarcastic
lebron best athlete of our generation if not all time basketball related I dont want to get into intersport debates...	Sarcastic

Fig 3. Results of sarcasm detection

In Fig 4 graph x-axis represents type of total sarcastic or non-sarcastic sentences and y-axis represents count of detected sentence type. Similarly we can put our own sentences in dataset.html file separated with comma where first value is then sentence id, second value is the sentence and third value is the emoticons.

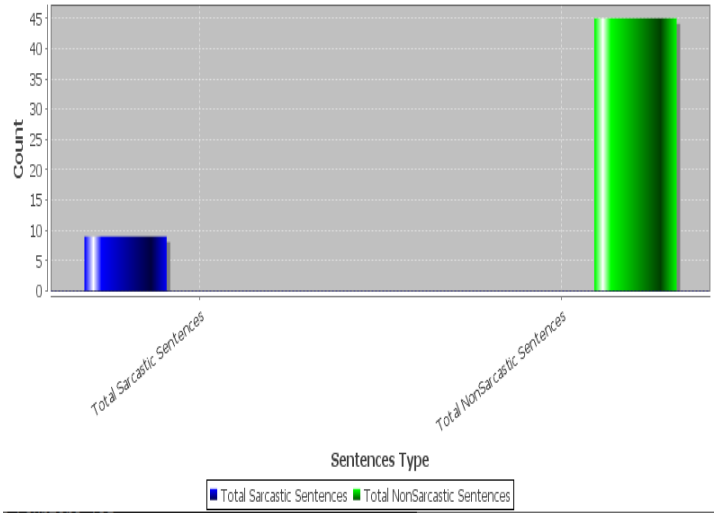

Fig 4. Graph of Total Sarcastic sentences vs Non Sarcastic Sentences

References

1. Albert Bifet and Eibe Frank. "Sentiment knowledge discovery in Twitter streaming data". In 13th International Conference On Discovery Science 2010, Springer, 1–15.
2. SK. Bharti, Bakhtyar Vachha, Ram Krushna Pradhan, K. Sathya Babu And S. K. Jena. "Sarcastic Sentiment Detection In Tweets Streamed In Real Time: A Big Data Approach". Digital Communications And Networks 2016, 2(3):108-12
3. Mondher Bouazizi And Tomoaki Otsuki. "A Pattern-Based Approach For Sarcasm Detection On Twitter". IEEE Access 2016, 4:5477-5488.
4. Koen Hallmann, Florian Kunzman, Christin Liebrecht, Antal Van Den Bosch And Margot Van Mulken. "Sarcastic Soulmates, Intimacy And Irony Markers In OSN Messaging". Linguistic Issues In Language Technology 2016, 14(7):1-23.
5. Dmitry Davidov, Oren Tsur And Ari Rappoport. "Semi-Supervised Recognition Of Sarcastic Sentences In Twitter And Amazon". 14th Conf. Comput. Natural Lang. Learn 2010,107-116.

6. Florian Kunneman, Christine Liebrecht, Margot Van Mulken, Antal Van Den Bosch. "Signaling Sarcasm: From Hyperbole To Hashtag". Information Processing & Management 2014, 51(4):500-509.
7. Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey And Pushpak Bhattacharyya. "Harnessing Cognitive Features For Sarcasm Detection". In The 54th Annual Meeting Of The Association For Computational Linguistics, 2017.
8. Aditya Joshi, Pushpak Bhattacharyya And Mark James Carman. "Automatic Sarcasm Detection: A Survey". Corr, Abs/1602.03426, 2016.
9. Liebrecht, C. C., Kunneman, F. A., And Bosh, A. P. J. "The Perfect Solution For Detecting Sarcasm In Tweets #Not". In Proc.WASSA, 2013. 29-37
10. Raquel Justo, Thomas Corcoran, Stephanie M. Lukin, Marilyn Walker, M. Ines Torres. "Extracting Relevant Knowledge For The Detection Of Sarcasm And Nastiness In The Social Web". Knowledge-Based Systems 2014, 69:124-133.
11. A. Bifet, E. Frank, Sentiment Knowledge Discovery In Twitter Streaming Data, In: 13th International Conference On Discovery Science, Springer, 2010.
12. Z. Tufekci, Big Questions For Social Media Big Data: Representativeness, Validity And Other Methodological Pitfalls, Arxiv Preprint Arxiv:1403.7400.
13. A.P. Shirahatti, N. Patil, D. Kubasad, A. Mujawar, Sentiment Analysis On Twitter Data Using Hadoop.
14. I. Ha, B. Back, B. Ahn, Mapreduce Functions To Analyse Sentiment Information From Social Big Data, Int. J. Distrib. Sens. Netw. 2015 (1) (2015) 1–11.
15. R.C. Taylor, An Overview Of The Hadoop/Mapreduce/Hbase Framework And Its Current Applications In Bioinformatics, BMC Bioinform. 11 (Suppl 12) (2010) 1–6.
16. R. González-Ibánez, S. Muresan, N. Wacholder, Identifying Sarcasm In Twitter: A Closer Look, In: Proceedings Of The 49th Annual Meeting On Human Language Technologies, ACL, 2011, Pp. 581–586.
17. D.Tayal,S.Yadav,K.Gupta,B.Rajput,K.Kumari,Polaritydetectionofsarcasticpolitical Tweets,In:Proceedingsofinternationalconferenceoncomputingforsustainable Globaldevelopment(Indiacom),IEEE,2014,Pp.625–628.
18. A.Rajadesingan,R.Zafarani,H.Liu,Sarcasmdetectionontwitter:Abehavioralmodeling Approach,In:Proceedingsoftheeighthacminternationalcon-Ference Onwebsearchanddatamining,ACM,2015,Pp.97–106.
19. R.J.Kreuz,G.M.Caucci,Lexicalinfluenceontheperceptionofsarcasm,In:Proceedings Oftheworkshoponcomputationalapproachestofigurative Language, ACL,2007,Pp.1–4.
20. C.Liebrecht,F.Kunneman,A.Vandenbosch,Theperfectsolutionfordetectingsarcasm Intweets#Not,In:Proceedingsofthe4thWorkshoponComputa-Tional Approachestosubjectivity,Sentimentandsocialmediaanalysis,ACL, Newbrunswick,NJ,2013,Pp.29–37.
21. E. Lunando, A. Purwarianti, Indonesian Social Media Sentiment Analysis With Sarcasm Detection, In: International Conference On Advanced Computer Sci- Ence And Information Systems (ICACSIS), IEEE, 2013, Pp. 195–198
22. P. Tungthamthiti, S. Kiyoaki, M. Mohd, Recognition Of Sarcasm In Tweets Based On Concept Level Sentiment Analysis And Supervised Learning Approaches, In: 28th Pacific Asia Conference On Language, Information And Computation, 2014, Pp. 404–413