International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

A Combinational Approach for Sarcasm Detection in Twitter

K. Karthika[1], M. Gayathridevi[2], M. Marikkannan[3]

[1, 2]M.E Computer Science and engineering, Institute of Road & Transport Technology (I.R.T.T), Erode Tamilnadu-638 316

[3]Assistant Professor (Senior), Department of Computer Science and Engineering, Institute of Road and Transport Technology (I.R.T.T), Erode Tamilnadu- - 638 316

Abstract: ***Sentiment analysis, also known as opinion mining is the qualitative method that uses natural language processing, text analysis and computational linguistics. The goal of sentiment analysis is to determine if a specific passage in the text shows positive, negative or neutral sentiment towards the subject. The main objective is to detect and analyze the emotional reactions of the speaker or writer based on his attitude. These emotions may change the polarity of the text and make it sarcastic. Due to these difficulties and the inherently tricky nature of sarcasm, it is generally ignored during social network analysis. As a result, the outcome of such analysis is affected adversely. Thus, sarcasm detection poses to be one of the most critical problems which we need to overcome while trying to yield high accuracy insights from abundantly available data. In this paper, we propose a combination of existing techniques to detect sarcasm on Twitter. It involves an effective preprocessing approach which are combined with four sets of features that can handle different types of sarcasm. We insist that the usage of semi-supervised learning method to train the sarcasm detection model which will increase the accuracy by approximately 3% than the state-of-art approaches. Hence, the proposed work may provide precision, recall and F-score around 90%, 96.5%, 92.0% respectively.***

Keywords: Sentiment analysis, Sarcasm Detection, Twitter, Machine learning

1. Introduction

Sarcasm is a superior kind of sentiment which shows an interfering factor that can toss the polarity of a given text. It is defined as "A form of verbal irony that is intended to express contempt or ridicule" by the Free Dictionary[1]. The Oxford dictionary defines sarcasm as, "The use of irony to mock or convey contempt". The fanciful nature of sarcasm makes it a challenge for sentiment analysis. A sarcastic sentence may carry a positive surface sentiment -for example, "Visiting the doctor for a shot, is so much fun!" or a negative surface sentiment -for example, "His performance in the IPL has been terrible!" or no surface sentiment -for example, the idiomatic expression "I am the KING of Arabia"

Sarcasm detection from text has now protracted to different data forms and methods. This Collaboration has resulted in motivating novelties for automatic sarcasm detection. To predict the correct sentiment of any text, identifying the sarcasm is vital. The challenges of sarcasm and the advantage of sarcasm detection to sentiment analysis have paved the way to an interest in automatic sarcasm detection as a research work. Automatic sarcasm detection refers to an approach that computationally predicts if a given text is sarcastic or not. Though this can be manually predicted easily, for automatic sarcasm detection, it's a big challenge. The difficulty is due to the nuanced ways in which sarcasm may be expressed. Recognizing sarcasm is one of the truly obscure tasks in natural language processing (NLP) too.

Recently, there is an evolving drift to use context beyond the text that is being classified. Like finding out sarcastic patterns and using them as features, contextual information, i.e., information beyond the target text, is also under the limelight. The proposed approach uses this feature as its semantic feature that deals with the sarcastic context.

TWITTER, one of the largest web source of the era for grabbing opinions and a perfect destine for the people to share their thoughts, spark a global conversation and convey real-time events. It claims to possess 330 million active users around the globe. The voluminous data generated from such a data ocean has been cherished by many companies and organization for improving their business. According to its official marketing site[2], Twitter states that it offers new marketing mix modelling insights and guidance on where to invest. It also utters that Twitter marketing campaigns can deliver 40% higher return on investment compared to other media channels. It assists the brand marketers in measuring the impact of their advertisements. Besides that, it shares the keys to In-Stream Video Ad effectiveness. The Evolving video consumption patterns represent a new opportunity for marketers. Mining valuable insights from twitter is not a straightforward task due to it noisy nature in terms of the tweets being posted. Adding to that sarcasm also poses to be a great challenge for deriving accurate results of the analysis. Table 1 shows a few sample tweets that are sarcastic.

1	"oh what a great invention-a store you can walk into and browse through books. Why didn't anyone do this before. #Sarcasm"
2	"Too much innovation! Can't handle it. #sarcasm"
3	"It's so nice to see my instructions were followed while I was off work 😊😊#sarcasm"
4	"No way. CNN is completely unbiased and objective in every way. #sarcasm"
5	"Education is so much fun"!! :P #sarcasm"
6	"Nokia deal is a good thing for our hardware partners #nokia #sarcasm"
7	"Good work #nokia. Was this the plan from day 1?#sarcasm"

Table 1: Sample Sarcastic Tweets

[1] https://www.thefreedictionary.com/

[2] https://marketing.twitter.com/na/en/insights.html

Hence the main goal of the proposed approach is to provide a better combination of techniques that would enhance the accuracy of the text-based analysis.

The rest of the paper is organized as follows: In Section 2, related work is presented. The proposed scheme and design goals in details are given in Section 3. Finally, we conclude in Section 5 along with future work.

2. Related Works

In the last few years, more attention has been given to Twitter sentiment analysis by researchers, and a number of recent papers have been addressed to the classification of tweets. However, the nature of the classification and the features used vary depending on the aim. In the previous works, researchers have used different features that include the hashtag, Lexical feature as uni-gram, bi-gram, tri-gram and n-gram, Pragmatic feature like emoticons, smiles, replies, etc, Intensifier, Interjection words like wow, oh, uh etc. Sarcasm in the text can be identified when text sentiment conflict with text situation or when text contradicts a fact or when a text contains sarcasm hashtag at the end.

In paper [1] Riloff et al. proposed a method to detect a specific type of sarcasm, where a positive sentiment contrasts with a negative situation. They introduced a bootstrapping algorithm that uses a collection of sarcastic tweets to automatically detect and learn expressions showing positive sentiment and phrases citing negative situations.

Bhat et al. [2] compared various classification algorithms such as Random Forest, Gradient Boosting, Decision Tree, Adaptive Boost, Logistic Regression and Gaussian Naïve Bayes to detect sarcasm in tweets from the Twitter Streaming API. The best classifier was chosen and paired with various pre-processing and filtering techniques using emoji and slang dictionary mapping to provide the best possible accuracy. The emoji and slang dictionary being the novel ideas introduced in this paper.

In paper [3] TomoakiOhtsuki et al, proposed a pattern-based approach to detect sarcasm on Twitter. The authors used four sets of features that cover the different types of sarcasm and then they used those to classify tweets as sarcastic and non-sarcastic. In particular, they emphasized the importance of pattern-based features for the detection of sarcastic statements.

NamrataBhan et al [4], projected a system that will measure sarcasm used in tweets. It used a score based algorithm to calculate the effect of sarcasm on texts. They compared the scores from different algorithms to present the most efficient way to detect sarcasm. The system also provided a separate portal to check the score of any sentence/text entered by the user and determine its score using the most accurate algorithm.

In paper [5] Barbieri et al. proposed a system that used seven sets of lexical features to detect sarcasm by its inner structure (for example unexpectedness, intensity of the terms or imbalance between registers), abstracting from the use of specific terms.

Bharati et al [6], suggested two approaches to detect sarcasm in the text of tweets. The first was a parsing-based lexicon generation algorithm (PBLGA) and the second was to detect sarcasm based on the occurrence of the interjection word.

In our work, we have insisted on Semi-supervised approach that learns sarcastic patterns.

3. Proposed System

The proposed flow of steps involves Extracting tweets, Preprocessing, Getting the emoticon and slang dictionary ready, Extracting features, Identifying the polarity& sarcasm, Training the model for Sentiment Classification. Finally validating the enhancement in accuracy that is being proposed by the model. These are depicted in Figure 1.

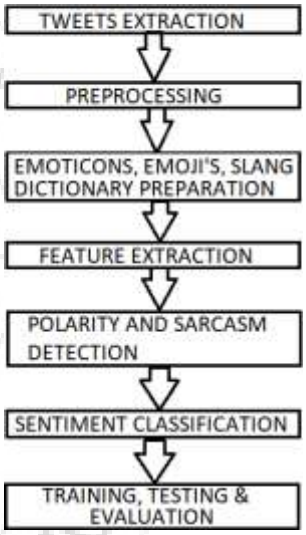

Figure 1: General block diagram

3.1 Tweets extraction

To collect tweets, Twitter's streaming API can be used. We can query the API for tweets containing the hashtag "#sarcasm". For extracting the required tweets, we need generate and get the Twitter API keys and connecting to Twitter Streaming APIs which gives access to all tweets as they are published. Then Read and save the Tweets in JSON format for preprocessing.

After this phase has been completed, we need to make a separate dictionary containing emoticons and slang words which is used to replace the expressions in tweets using them.

3.2 Preprocessing

The Preprocessing steps are shown in figure 2.

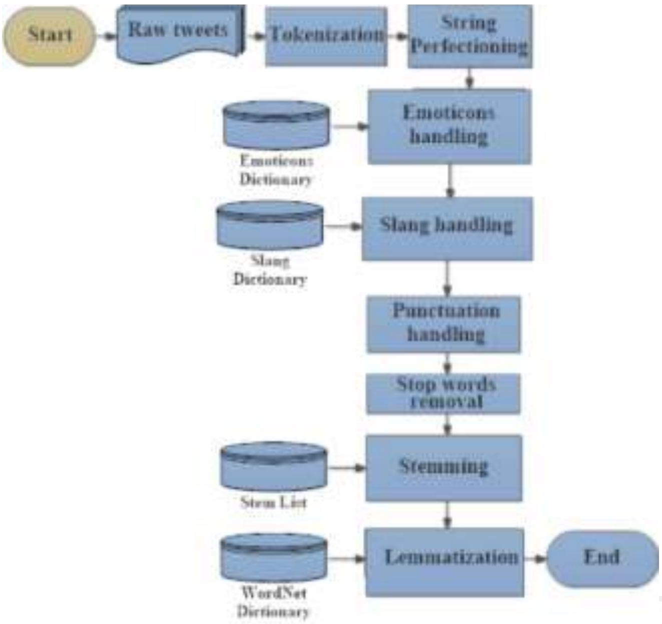

Figure 2: Flow chart for Preprocessing

Following is a detailed explanation of the pre-processing techniques.

3.2.1 Raw tweets

The Tweets as such are very noisy in nature. The contain multiple languages, hyperlinks, images etc. we need to extract only the tweet statements after removing all unwanted tags in the raw JSON format.

3.2.2 Tokenization

Tokenization is the process of breaking a sentence into words, phrases, symbols or other meaningful tokens.

Example of tokenization - The sentence- "What an awesome day, it is today @ office!" will be broken into tokens 'what', 'an' 'awesome', 'day', 'it', 'is', 'today', 'at', '@', 'office', '!' after tokenization.

3.2.3 String perfectioning

The tokenized strings of the tweets may contain few elongated words that stress on the emotions. The words that are elongated to express a particular emotion strongly and the words with wrong spellings are also to be corrected and replaced with their original words.

3.2.4 Emotion Handling

The tokens have to be compared with the emoticons dictionary elements. If match found, they are changed to the corresponding meanings. The link[3] provides a dictionary for the emoticons.

3.2.5 Slang Handling

The Twitter users vary in different aspects like age, demography, language etc. Each have their own habit of text shortening slangs. These have to be replaced with original words as they make the process of analysis easier.
Example: 'gr8' had to be replaced with 'great'.

[3] https://www.csh.rit.edu/~kenny/misc/smiley.html

3.2.6 Punctuation Handling

This includes apostrophe handling like 'n't' has to be replaced with 'not'. Once the apostrophes are handled, all the remaining punctuations and numbers are to be removed from the text.

3.2.7 Stop words Handling

These are the words that do not carry any meaning. Example: 'a', 'is', 'the' etc. These have to be removed to get those tokens that are meaningful.

3.2.8 Stemming

It is the process of finding the root word of a specified token. It removes various suffixes. A predominantly used method is the Porter stemming which involves removing the common morphological and in-flexional endings from words in English.

3.2.9 Lemmatization

Lemmatization is the process of analyzing the words with the use of vocabulary and returning the dictionary form of a word.
WordNet dictionary is mostly used for this purpose.

3.3 Emoticon, emoji's and Slang Dictionary

Each of them have to be replaced before sending the tweets for sentiment analysis. The following examples depict how the emoticons and slang words are replaced. Example: ":d" => "good", "yaaay" => "good", ":(" => "bad".

3.4 Feature extraction

The features used for sarcasm detection is presented as a chart in figure 3.

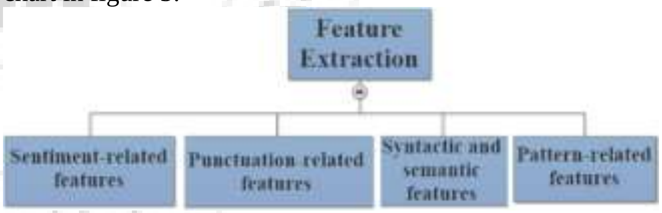

Figure 3: Feature extraction chart

3.4.1 Sentiment-related features

This maintains two lists of different words that are qualified as "positive words" and "negative words". First list contains words that have positive emotional content (e.g., "love", "happy", etc.) and the second list contains negative emotional content (e.g., "hate", "sad", etc.). Using these two lists, two features namely "Positive words (pw)" and "Negative words (nw)" are to be extracted by counting the number of positive and negative words respectively in the tweet. Verbs, adverbs and adjectives have higher emotional content than nouns; therefore positive and negative words that have the associated PoS-tag, are to be counted again and used to create two more features that we denote PW and NW and which denotes the number of highly emotional positive words and highly emotional negative words. It also takes emoticons into consideration. Ratio of emotional words is calculated as in (1).

$$\rho(t) = \frac{(\delta \cdot PW + pw) - (\delta \cdot NW + nw)}{(\delta \cdot PW + pw) + (\delta \cdot NW + nw)} \quad (1)$$

Where t is the tweet, pw, PW, nw and NW denote respectively the number of positive words (other than highly emotional ones), that of highly emotional positive words, that of negative words (other than highly emotional ones) and that of highly emotional words. Smiley is a weight bigger than 1 given to the highly emotional words. In case the tweet does not contain any emotional word, is set to 0.

3.4.2 Punctuation-related features

For each tweet, the number of exclamation marks, question marks, dots, all-capital words, quotes are to be counted. Another feature is also included by checking if any of the words contain a vowel that is repeated more than 3-4 times (e.g."looooove"). If such a case is found then the feature is set to be true else it is set to false.

3.4.3 Syntactic and sematic features:

Along with the punctuation-related features, some common expressions are used usually in a sarcastic context. It is possible to correlate these expressions with the punctuation to decide whether what is said is sarcastic or not. Besides, in other cases, people tend to make complicated sentences or use uncommon words to make it ambiguous to the listener/reader to get a clear answer. The features like Count of uncommon words, Presence of sarcastic text or expression, number of interjections, number of laughter's.

3.4.4 Pattern-related features

Here the words are to be divided into 2 classes, CT and GFI. CT includes the words whose content is more important and GFI includes the words whose grammatical function is more important as depicted by (2).

$$\mathbf{Lmin} \leq \mathbf{Length(pattern)} \leq \mathbf{Lmax} \quad (2)$$

Where Lmin and Lmax represent the minimal and maximal allowed length of patterns in words and Length (pattern) is the length of the pattern in words. The number of pattern lengths is

NL = (Lmax - Lmin + 1). We classify the resulted patterns into NF sets where, NF = NL × NS .Next, resemblance, res(p,t), is to be calculated as in (3).

$$res(p,t) = \begin{cases} 1, & \text{if the tweet vector contains the pattern as it is, in the same order} \\ \alpha \cdot n/N, & \text{if n words out of the N words of the pattern appear} \\ 0, & \text{if no word of the pattern appear in the tweet} \end{cases} \quad (3)$$

3.5 Polarity and Sarcasm Detection

There are three model entities explained in figure 4.

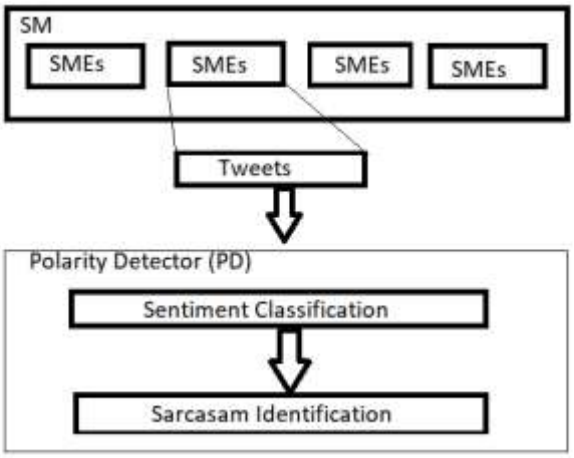

Figure 4: System model for opinion mining

Social media (SM): It is a social networking website (Facebook, Twitter, Amazon, etc.), where people use to write an opinion, reviews, and post some blogs, or micro-blogs about any social media entities.

Social media entities (SME): An entity about which, users want to post and retrieve comments, tweets and give reviews on social media.

Polarity detector (PD): An automated system which is capable to identify the actual sentiment or sarcasm sentiment from text.

Whenever an event starts or a product is launched, people start writing tweets, reviews, comments etc. on the social media. People rely very much on these social media to get the exact review about any product before they buy. Organizations also dependent on these sites to know the marketing status of their product and subsequently improve their product quality. However, finding and monitoring correct opinions or reviews about SMEs remain a formidable task. It makes difficult for humans to read all the reviews and decode the sarcastic opinions. In addition, the average human reader will have difficulty in recognition of sarcasm in twitter data, product review, etc., which misleads the individuals or organizations.

To avoid this difficulty, there is a need for an automated sarcasm recognition system. Figure 4 shows how to find opinions automatically from SMEs. Tweet is one of the SMEs which pass through a polarity detector to get the opinions about a given tweet. Polarity detector classify it into either negative, positive or neutral. If the tweet is classified as either positive or negative than further checks are required to see whether, it has actual positive/negative sentiment or sarcastic sentiment. Based on the result of the polarity detector, users can take the decision.

3.6 Sentiment Classification

It is the process of determining if the sentiment of a particular text is positive, negative or neutral) can be performed using SentiWordNet or TextBlob.
SentiWordNet is a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet three

sentiment scores: positivity, negativity, objectivity. TextBlob is a Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. The disintegrated structure of sarcasm detection is depicted in figure 5.

Figure 5: Disintegrated structure of sarcasm detection

3.7 Training, Testing and Evaluation

For training the model for classification “Random Forest”, “Linear SVC (support vector clustering)” and “Gradient Boost” is expected to act as better classifier. To compare the results with existing work, three parameters are considered, namely, precision (4), recall (5) and f-score (6). Precision is a statistical parameter that shows, how many tweets correctly identified out of total identified tweets by algorithm with the reference of ground truth. Correctly identified tweets are called true positive (Tp) as algorithm and ground truth, both identified positive and the tweets are incorrectly identified called false positive (Fp) as algorithm identified positive but ground truth was negative. Similarly, how many of tweets correctly rejected is called true negative (Tn) as both algorithm and ground truth identified negative and how many tweets incorrectly rejected called false negative (Fn) as algorithm identified negative but ground truth was positive.

$$Precision = \frac{T_p}{T_p + F_p} \tag{4}$$

$$Recall = \frac{T_p}{T_p + F_n} \tag{5}$$

$$F - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{6}$$

Where, Tp = True positive, Fp = False positive, Fn = False negative. These values of the proposed approach, when compared with the values from table (2) may yield around 90% precision, 96.5% recall and around 92.0% F-score. The accuracy may improve by at least 3%.

Table 2: Precision, recall and F-score of state-of-art approaches

Approach	Precision	Recall	F-score
Riloff et al. [26]	65.0%	40.8%	50.1%
Bharti et al. [4]	89.0%	96.0%	90.0%
Barbieri et al. [22]	88.0%	87.0%	88.0%

4. Conclusion and Future Works

Sarcasm analysis is one of most challenging task as it doesn’t have any pre-defined structure. In this paper, we have proposed a way of improving the existing sarcasm detection techniques by including better pre-processing and text mining techniques such as emoji and slang detection. For classifying tweets as sarcastic and non-sarcastic there are various techniques used. Our approach is expected to give better accuracy than the state of art works. It may provide around 90% precision, 96.5% recall and around 92.0% F-score. The accuracy may improve by at least 3%.

In a future, we will study and implement how to use the output of the current findings to enhance the performances of sentiment analysis. More over Sarcasm detection in non-“#sarcasm” hash tag tweets, is much more challenging than those tweets with #sarcasm hash tag. So, we will also take a deep dive into it and unleash the patterns in such tweets.

References

[1] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, ‘‘Sarcasm as contrast between a positive sentiment and negative situation,’’ in Proc. Conf. Empirical Methods Natural Lang. Process., Oct. 2013, pp. 704–714.

[2] Anukarsh G Prasad; Sanjana S, Skanda M Bhat, B S Harish, “Sentiment Analysis for Sarcasm Detection on Streaming Short Text Data”(2017 2nd International Conference on Knowledge Engineering and Applications IEEE)

[3] MondherBouazizi, TomoakiOhtsuki, “A Pattern-Based Approach for Sarcasm Detection on Twitter” (2016 IEEE Access)

[4] NamrataBhan, Prof. Mitchell D’silva, “Sarcasmometer using Sentiment Analysis and Topic Modeling” (2017 IEEE Access)

[5] F. Barbieri, H. Saggion, and F. Ronzano, “Modelling sarcasm in twitter, a novel approach,” in Association for Computational Linguistics, pp. 50–58, 2014.

[6] Santosh Kumar Bharti, KorraSathyaBabu, Sanjay Kumar Jena, “Parsing-based Sarcasm Sentiment Recognition in Twitter Data” (IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining), pp. 1373-1380, 2015

Author Profile

K.Karthika received the B.E degree in Computer Science & Engineering from Velammal Engineering College, Anna University, Chennai in 2010. She is an Oracle certified Associate who has worked for companies like TCS, Chennai and Yuvasys Technologies India Private Ltd, Coimbatore. She is currently pursuing her Master’s degree in Computer Science and Engineering at Institute Of Road And Transport Technology, Erode, affiliated to Anna University.

M.Gayathridevi received the B.Tech degree in Information Technology from Institute Of Road and Transport Technology, Erode, affiliated to Anna University in 2017. She is currently pursuing her

Master's degree in Computer Science and Engineering at Institute Of Road And Transport Technology, Erode, affiliated to Anna University.

M. Marikkannan received the B.E Computer Engineering Degree from Government college of Engineering, Tirunelveli Manonmanium Sundaranar University, M.E Computer Science and Engineering from College of engineering, Guindy, Anna University, Ph.D Degree from Anna University. He is currently working as Senior Assistant Professor, Department of computer Science and Engineering at Institute Of Road and Transport Technology, Erode, affiliated to Anna University since 2001.