2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Parsing-based Sarcasm Sentiment Recognition in Twitter Data

Santosh Kumar Bharti
Computer Science Engineering
National Institute of Technology
Rourkela, Orissa, India- 769008
Email: 513cs1037@nitrkl.ac.in

Korra Sathya Babu
Computer Science Engineering
National Institute of Technology
Rourkela, Orissa, India- 769008
Email: ksathyababu@nitrkl.ac.in

Sanjay Kumar Jena
Computer Science Engineering
National Institute of Technology
Rourkela, Orissa, India- 769008
Email: skjena@nitrkl.ac.in

***Abstract*—Sentiment Analysis is a technique to identify people's opinion, attitude, sentiment, and emotion towards any specific target such as individuals, events, topics, product, organizations, services etc. Sarcasm is a special kind of sentiment that comprise of words which mean the opposite of what you really want to say (especially in order to insult or wit someone, to show irritation, or to be funny). People often expressed it verbally through the use of heavy tonal stress and certain gestural clues like rolling of the eyes. These tonal and gestural clues are obviously not available for expressing sarcasm in text, making its detection reliant upon other factors. In this paper, two approaches to detect sarcasm in the text of Twitter data were proposed. The first is a parsing-based lexicon generation algorithm (PBLGA) and the second was to detect sarcasm based on the occurrence of the interjection word. The combination of two approaches is also shown and compared with the existing state-of-the-art approach to detect sarcasm. First approach attains a 0.89, 0.81 and 0.84 $precision$, $recall$ and $f-score$ respectively. Second approach attains 0.85, 0.96 and 0.90 $precision$, $recall$ and $f-score$ respectively in tweets with sarcastic hashtag.**

***Keywords*—Natural language processing, Opinion mining, Parsing, Sarcasm, Sentiment, Tweets.**

I. Introduction

Several trends are opening up the era of sentiment analysis, that analyse ones' attitude and opinion in social media, including Facebook, Twitter, blogs, etc. It becomes a much known way to infer data. The main aim of sentiment analysis is to identify the overall orientation (positive, negative, or neutral) in a given piece of text. Sarcasm is a special type of sentiment which plays a role as an interfering factor that can flip the polarity of the given text. Example: "Nothing I love more than a crowded library with no seats" :-) $\#sarcasm$. In this example, people use to express the positive sentiment (love) but overall tweet reflect negative sentiment toward the library. Unlike a simple negation, a sarcastic text, typically conveys a negative opinion using only positive words or even intensified positive words. Recognition of sarcasm is one of the most difficult tasks in natural language processing. The average human reader will have difficulty in recognition of sarcasm in twitter data, product review, blogs, online discussion forum, etc. Sarcasm is defined by various dictionaries in their own style. According to Macmillan English dictionary, sarcasm is "the activity of saying or writing the opposite of what you mean, or of speaking in a way intended to make someone else feel stupid or show them that you are angry". The Random House dictionary defines sarcasm as "a harsh or bitter derision or irony" or "a sharply ironical taunt; sneering or cutting remark". The Collins English dictionary defines it as "mocking, contemptuous, or ironic language intended to convey insults or scorn". Another definition of sarcasm, by Merriam-Webster is "a mode of satirical wit depending for its effect on bitter, caustic, and often ironic language that is usually directed against an individual". Twitter is an online social networking service that allows users to post and read short messages, called "tweets". Twitter allow users to write short messages, i.e. 140 characters per tweet. Due to this limitation, user often uses symbolic and figurative text in their message such as @USER, smiles, emoticons, punctuation mark and interjections. In addition, users usually post a lot of tweets in complex sentence structures, that sense difficult to identify for a machine.

The analysis of existing research concludes that, there are some features present in the text, that plays a vital role to identify sarcasm. They are hyperbole, pragmatic, lexical, etc. [1]–[3]. A hyperbolic text is highly probable to classify sarcastic as one can analyse sarcastic utterance with a hyperbole. For example: 'fantastic weather' when it rains, is identified as sarcastic with more ease than a sarcastic utterance without a hyperbole ('the weather is good' when it rains). Combination of features like interjection, intensifier, quotes and punctuation mark in a given text is called hyperbole [3]. The presence of intensifier in a sentence increases the probability of sarcasm. In a given text adjective and adverb act as an intensifier. Few interjection words that are frequently used in a text, as wow, yay, yeah, oh, nah, aha, etc. There are also some punctuation marks and quotes such as ?????, !!!!!, ' ', " " that are commonly used in texts. The lexical features $viz.$ uni-gram, bi-gram, tri-gram and n-gram are useful to identify sarcasm [4]. Another, feature which is useful to identify sarcasm is pragmatics [2] such as smiles and emoticons. In the realm of Twitter, we observed that most of the sarcastic tweets that start with the interjection word [5] have a high tendency to classify as sarcastic. Few examples are shown in TABLE I and the interjection words are marked as bold.

After analysis, we conclude that if any tweet starts with interjection word and immediately followed by either adjective or adverb (marked as underlined italic), it can be classified as sarcastic (shown in tweet numbers 5, 6, 7 and 8). Similarly, if any tweet starts with interjection word and it contains adverb and immediately followed by adjective or vice-versa (marked as underlined italic), can also be classified as sarcastic (shown

ASONAM '15, August 25-28, 2015, Paris, France

DOI: http://dx.doi.org/10.1145/2808797.2808910

TABLE I: Sample tweets starting with interjection word

1	**Wow**, you beat a 3 seed that's 4 inches shorter than you. You are just *so amazing* #sarcasm
2	**Wow**, that's a *huge discount*, Im not buying anything!! #sarcasm
3	**Oh** wow look at the *most realistic* doughnuts in a video game #sarcasm
4	**Wow**, what an *amazing night* this has turned out to be #sarcasm
5	**Wow**, *now* that I know about Photoshop, my decade-long eating disorder is cured! #sarcasm
6	**Wow**, *so* excited to see all kinds of couple pictures on Saturday and all the cute things they get for each other #sarcasm
7	**Wow**, *publicly* promoting "The 1%" concept, great! #sarcasm
8	**Aha**, *great* night #sarcasm

in tweet numbers 1 and 3). In any tweet that starts with interjection word and contains either structure first adjective followed by a noun or first adverb followed by a verb (underlined italics) can also be classified as sarcastic (shown in tweet numbers 2 and 4). Further, a new type of sarcasm text that having contradiction between negative sentiments and positive situation, such as, "I **hate** Australia in cricket, because they *always win*". $\#sarcasm$. This tweet looks negative as hate (negative sentiment) is presented in bold, but situation tells positive as they always win marked as underlined italic.

The rest of the paper is organized as follows: In Section II, related work is presented. Preliminary is given in Section III. The proposed scheme, design goals and implementation details are given in Section IV. Experimental analysis is shown in Section V. Finally, we conclude in Section VI.

II. Related Work

In recent times, research interest grew rapidly towards sarcasm detection in text [1]–[16]. Many researchers have investigated sarcasm on the data collected from various sources such as tweets on Twitter, Amazon product reviews, website comments, Google books and online discussion forums with the help of various features such as lexical, pragmatics and hyperbole. Sarcasm detection can be classified into three categories, namely lexical, pragmatics and hyperbole.

Lexical feature includes text properties such as uni-gram, bi-gram, tri-gram and n-gram. Kreuz *et al.* [17] introduced the lexical feature, that play a vital role to recognize irony and sarcasm in text. Utsumi *et al.* [18] suggested that extreme adjectives and adverbs often provide an implicit way to display negative attitudes $i.e.$ sarcasm. Kreuz in his subsequent work along with Caucci [2] used these lexical and syntactic features to recognize sarcasm in text and also discussed the role of different lexical factors, such as an interjection (e.g., gee or gosh) and punctuation symbols (e.g., ?). A semi-supervised approach was introduced by Davidov *et al.* [13] to detect sarcasm in tweets and Amazon product reviews. They used two interesting lexical features, namely pattern-based (high frequency words and content words) and punctuation-based to build a weighted k-nearest neighbor classification model to perform sarcasm detection. Gonzlez-Ibnez *et al.* [1] explored numerous lexical features (derived from LWIC [19] and WordNet affect [20]) to identify sarcasm. Riloff *et al.* [4] used a well-constructed lexicon based approach to detect sarcasm based on an assumption that sarcastic tweets are a contrast between a positive sentiment and a negative situation. For lexicon generation, they used uni-gram, bi-gram and tri-gram features.

Pragmatic features include symbolic or figurative texts like smilies, emoticons, replies. Kreuz *et al.* [17] introduced the concept of pragmatic features for the first time. This feature is very much helpful to identify sarcasm in textual data. Gonzlez-Ibnez *et al.* [1] explored it further with emoticons, smiles and replies. They developed a system using the pragmatics features to identify sarcasm in tweets. Tayal *et al.* [14] used pragmatic features to identify sarcasm in political tweets. Rajadesingan *et al.* [15] used a systematic approach for sarcasm detection in tweets and used psychological and behavioural pragmatic features of an user with their present and past tweets.

TABLE II: Types of sarcasm occur in text

T1	Contrast between positive sentiment and negative situation
T2	Contrast between negative sentiment and positive situation
T3	Fact Negation - text contradicting a fact
T4	Likes and Dislikes Prediction – behaviour based
T5	Lexical Analysis - sarcasm hashtag based
T6	Temporal Knowledge Extraction - tweets contradicting facts about event

TABLE III: Types of feature

F1	Lexical- unigram, bi-gram, tri-gram, n-gram, #hashtag
F2	Pragmatic- smiles, emoticons, replies
F3	Interjection- yay, oh, wow, yeah, nah, aha, etc.
F4	Intensifier- adverb, adjectives
F5	Punctuation Mark- !!!!!, ????
F6	Quotes- " " , ' '
F7	Pattern based- high frequency words and content words

TABLE IV: Types of domain

D1	Tweets of Twitter
D2	Online product reviews
D3	website comments
D4	Google Books
D5	Online Discussion Forums

Hyperbole plays the most important role in order to identify sarcasm in the text. A combination of the properties such as intensifier, interjection, quotes, punctuations etc. in the text is called hyperbole. Lunando *et al.* [5], focused only on interjection words such as aha, bah, nah, wah, wow, yay, uh, etc. for sarcasm identification. They conclude that, if the text contains interjection words, it has more tendency to be classified as sarcastic. Liebrecht *et al.* [3] focused on hyperbole to detect sarcasm in tweets as utterance with a hyperbole ('fantastic weather' when it rains) is identified as sarcastic with more ease than the utterance without a hyperbole ('the weather is good' when it rains). The utterance with the hyperbolic 'fantastic' may be easier to interpret more sarcastic than the utterance with the non-hyperbolic 'good'. Filatova *et al.* [8] focused on hyperbole features to identify sarcasm in document level text as only a phrase or a sentence is not sufficient for sarcasm detection. They considered the context of a sentence and the surrounding sentences to improve the accuracy of the detection. For this experiment, regular and sarcastic Amazon product reviews were used. Tungthamthiti *et al.* [16], explored concept level knowledge to identify an indirect contradiction between sentiment and situation using the hyperbole feature as intensifier. According to them words like raining, bad weather is conceptually same. So, if raining is present in tweet then consider it as a negative situation. They also focus on coherency if tweet contains multiple sentences. Peng *et al.* [11], presented a multi-strategy ensemble classification algorithm and a comprehensive sarcasm feature set including lexical, syntactic, semantics and constructions for automatic detection of sarcasm in both English and Chinese social media.

There are five types of sarcasm occurring in the text. They are (i) when text sentiment conflict with text situation. (ii) when text contradicts a fact. (iii) when the authors likes and dislikes conflict with the text. (iv) when a text contains sarcasm hashtag at the end. (v) when a text conflict with the fact about any event such as a festival, birthday, sports, etc. A summary of the types is shown in TABLE II. Features play an important role in the detection of sarcasm in the text. In the literature, researchers used different features. A list of these features is given in TABLE III. These are (i) Hashtag, if any text contains hashtag sarcasm, then its easy to identify the text as sarcastic. (ii) There are some lexical feature as uni-gram, bi-gram, tri-gram and n-gram through which one can easily identify sarcasm in text. Ex: bi-gram "amazing night" people often say this phrase sarcastically. (iii) Pragmatic feature like emoticons, smiles, replies, etc. These features often used by the author to write sarcastic text. (iv) Another feature which widely used in sarcastic text is an intensifier. Adverbs and adjectives are often used as intensifiers such as fantastic weather, so pleased, etc. (v) There are some interjection words like wow, oh, uh etc. which are used highly in sarcastic texts. Similarly, punctuation mark, quotes, etc. play big role in identification of sarcasm. (vi) There are some high frequency words which also play major role in recognition of sarcasm in text. Ongoing research on sarcasm detection, researchers used data in various domains. A list of these domains is given in TABLE IV. They are tweets, product reviews, online discussion forum, google books, and website comments. A consolidated table showing features available, type of sarcasms and domains that are deployed are shown in TABLE V. The next section details about the preliminaries of this article.

III. PRELIMINARIES

Opinions posted in social media can be classified as positive, negative and neutral. A positive sentiment can be further classified as actual positive or sarcastic sentiment. Similarly, negative sentiment can be actual negative or sarcastic.

With the explosive growth of social media on the web, individuals and organizations are increasingly using this content in decision making [21]. A system model that shows, how the social media is useful for user in their decision making is shown in Fig. 1. There are three model entities:

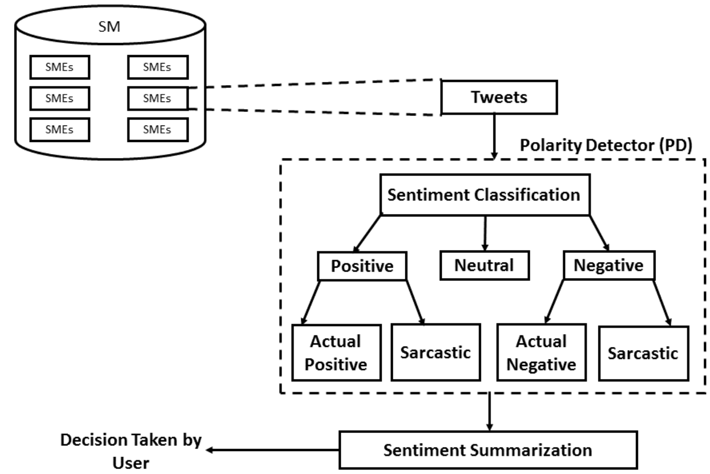

Fig. 1: System model for decision making from users' opinions.

- *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):* A 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. Fig. 1 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.

TABLE V: Previous studies in sarcasm detection in text.

Study	Types of Sarcasm						Types of feature							Domain				
	T1	T2	T3	T4	T5	T6	F1	F2	F3	F4	F5	F6	F7	D 1	D2	D3	D4	D5
Kreuz *et al.*, 1995					✓		✓	✓		✓				✓				
Utsumi *et al.*, 2000					✓					✓				✓				
Kreuz *et al.*, 2007					✓		✓		✓		✓			✓				
Tsur *et al.*, 2010	✓			✓	✓		✓				✓		✓		✓			
Davidov *et al.*, 2010					✓		✓				✓			✓	✓			
Gonzlez-Ibnez *et al.*, 2011					✓		✓	✓						✓				
Filatova *et al.*, 2012			✓	✓				✓	✓		✓	✓			✓			
Riloff *et al.*, 2013	✓				✓		✓							✓				
Lunando *et al.*, 2013					✓				✓					✓				
Liebrecht *et al.*, 2013	✓				✓		✓			✓	✓			✓				
Rajadesingan *et al.*, 2014	✓			✓	✓		✓	✓						✓				
Tungthamthiti *et al.*, 2014	✓				✓		✓			✓				✓				
Peng *et al.*, 2014	✓				✓		✓		✓				✓	✓				
Raquel *et al.*, 2014					✓			✓					✓	✓	✓	✓		✓
Kunneman *et al.*, 2014					✓		✓		✓	✓	✓			✓				
Barbieri *et al.* 2014					✓		✓							✓				
Tayal *et al.*, 2014					✓		✓	✓				✓		✓				
Nitin *et al.*, 2014			✓	✓	✓	✓	✓							✓				
Pielage *et al.*, 2014					✓		✓	✓						✓	✓	✓	✓	✓

A. *Part-of-speech (POS) Tagging*

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. 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.

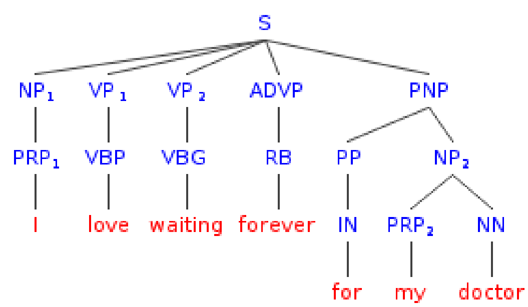

Fig. 2: Parse tree for tweet : I love waiting forever for my doctor

B. *Parsing and Parse Tree*

Parsing is the process of analysing the grammatical structure of a language. Given a sequence of words, a parser forms units like subject, verb, object and determines the relations between these units, according to the rules of a formal grammar and generate a parse tree. Fig. 2 and Fig. 3 depict the examples of the parse trees for tweets "I love waiting forever for my doctor" and "I hate Australia in cricket, because they always win. The notations S, NP, VP, ADJP, ADVP denotes statement, noun phrase, verb phrase, adjective phrase and adverb phrase respectively. To generate a parse tree, one requires the paring data. Here, TEXTBLOB was used to find parse data of tweets. The output given by TEXTBLOB for the above tweet (I love waiting forever for my doctor) is I/PRP/B-NP/O, love/NN/I-NP/O, waiting/VBG/B-VP/O, forever/RB/B-ADVP/O, for/IN/B-PP/B-PNP, my/PRP$/B-NP/I-PNP, doctor/NN/I-NP/I-PNP). A parse tree as shown in Fig. 2 was generated by these tags. It shows the tweet type (T1) mentioned in TABLE III as "I love" indicating positive sentiment phrase and "waiting forever" indicating negative situation phrase, whereas, Fig. 3 shows the tweet type (T2) mentioned in TABLE III as " hate" indicating negative sentiment and "always win" indicating a positive situation.

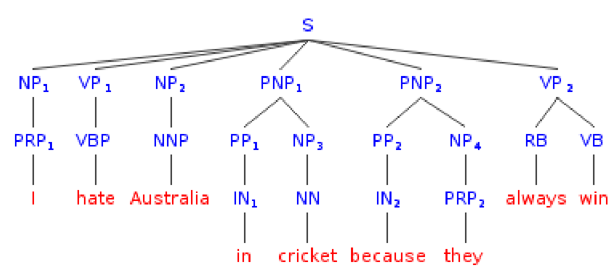

Fig. 3: Parse tree for : I hate Australia in cricket, because they always win.

IV. Proposed Scheme

This section deals with the design goals and proposed algorithms. In all of the previous work for identifying sarcasm, efforts were made to design solutions that meet the requirements of various types of sarcasm as discussed in TABLE III and different feature set shown in TABLE IV. We observed that the authors have not yet worked on T2 type of sarcasm and hyperbolic text that starts with interjection words (only discussed by Lunando *et al.* [5] for Indonesian social media). Based on these two limitations, two proposals are made.

1) A parsing based lexical generation algorithm (PBLGA) for sarcasm identification in Twitter data is required to recognize sarcasm in both forms of tweets: (a) contradiction of negative sentiment and positive situation (b) contradiction of positive sentiment and negative situation.
2) An `Interjection_word_start` (IWS) algorithm is given to identify sarcasm in tweets that start with interjection words like wow, yay, etc.

A. PBLGA

PBLGA generates the lexicon to identify sarcasm in tweets. Where, tweet‘s sentiment contradicts with the tweet‘s situation. Generated lexicon contains phrase in four categories, namely positive sentiment, negative situation, negative sentiment and positive situation. PBLGA algorithm is given below.

Algorithm 1: PBLGA

Input: Tweet corpus (C)

Output: Lexicon for negative sentiment, positive sentiment, negative situation and positive situation.

Notation: VP: verb phrase, NP: noun phrase, $ADJP$: adjective phrase, $ADVP$: adverb phrase, PF: parse file, TWP: tweet wise parse, SF: sentiment file, sf:situation file, PSF: positive sentiment file, NSF: Negative sentiment file, psf: positive situation file, nsf: negative situation file, SC: sentiment score, P: phrase, T: Tweet.

Initialisation **:** $SF = \{ \phi \}$, $sf = \{ \phi \}$, $PSF = \{ \phi \}$, $NSF = \{ \phi \}$, $psf = \{ \phi \}$, $nsf = \{ \phi \}$,

1: **for** T in C **do**
2: k = `find_parse` (T)
3: $PF \leftarrow TF \cup k$
4: **end for**
5: **for** TWP in PF **do**
6: k = `find_subset` (TWP)
7: **if** $k = NP \parallel ADJP \parallel (NP + VP)$ **then**
8: $SF \leftarrow SF \cup k$
9: **else if** $k = VP \parallel (ADVP + VP) \parallel (VP + ADVP) \parallel (ADJP + VP) \parallel (VP + NP) \parallel (VP + ADVP + ADJP) \parallel (VP+ADJP+NP) \parallel (ADVP+ADJP+NP)$ **then**
10: $sf \leftarrow sf \cup k$
11: **end if**
12: **end for**
13: **for** P in SF **do**
14: SC = `sentiment_score` (P)
15: **if** SC >0.0 **then**
16: $PSF \leftarrow PSF \cup P$
17: **else if** SC <0.0 **then**
18: $NSF \leftarrow NSF \cup P$
19: **else**
20: Neutral Sentiment Phrase
21: **end if**
22: **end for**
23: **for** P in sf **do**
24: SC = `sentiment_score` (P)
25: **if** SC >0.0 **then**
26: $psf \leftarrow psf \cup P$
27: **else if** SC <0.0 **then**
28: $nsf \leftarrow nsf \cup P$
29: **else**
30: Neutral Situation Phrase
31: **end if**
32: **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, step 2 finds parse of tweets in corpus and store it tweet wise (parse of one tweet in one line and next tweet in next line) in parse file. Step 6 finds the tweet wise subset in parse file. If a subset of parse contains either noun phrase followed by verb phrase or adjective phrase or only noun phrase, then add the phrase into the sentiment file. In the way of learning sentiment phrase, singular proper noun (NNS) and plural proper noun (NNPS) are not considered as it doesn't contain any sentiment. Similarly, in the verb phrase, gerund or present participle Verb (VBG) are not considered. In order to learn situation phrase, there is no such restriction. If subset of parse tree contains either (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 adverb phrase followed by adjective phrase) or (verb phrase followed by adjective phrase followed by noun phrase) or (adverb phrase followed by adjective phrase followed by noun phrase) then the algorithm add the phrase into situation file as shown in Algorithm 1.

To find the sentiment score of each generated phrase, apply sentiment score criteria to separate negative sentiment and positive sentiment from sentiment file. Similarly, find negative situation and positive situation from situation file. To find sentiment score, use Equations 1, 2 and 3 as follows:

$$PR = \frac{PWP}{TWP} \tag{1}$$

$$NR = \frac{NWP}{TWP} \tag{2}$$

$$SentimentScore = PR - NR \tag{3}$$

where,
PR= Positive Ratio, NR= Negative Ratio, PWP= Number of Positive words in a given phrase, NWP= Number of

Negative words in a given phrase, TWP= Total words in given phrase

Equations 1 and 2 gives a positive ratio and negative ratio for the given phrase respectively. Finally, Equation 3 gives the sentiment score of that phrase. If the sentiment score is greater than 0.0 in sentiment file, algorithm add it in a positive sentiment file otherwise in negative sentiment file. Similarly, if the sentiment score is greater than 0.0 in situation file, algorithm add it in the positive situation file otherwise in the negative situation file. If the sentiment score is exactly 0.0 or phrase doesn't contained any sentiment than we decide manually based on text context for both sentiment and situation file. Finally, we use the learned lists of sentiment and situation phrases to recognize sarcasm in new tweets by identifying contexts, that contain a positive sentiment in close proximity to a negative situation phrase.

B. IWS

In the second approach (Algorithm 2), a hyperbole feature considered to identify sarcasm in tweets that starts with interjection word such as oh, wow, yay, yeah, nah, aha, uh, etc. Hyperbole is a combination of interjection and intensifier [2], [3] as already discussed in TABLE I. If interjection appears in the beginning of the tweet and intensifier appears other than in the beginning, Than a tweet has high probability to classify as sarcastic. Presence of adjective or adverb makes a text intensified as (Fantastic weather), Here, Fantastic is an adjective acts as an intensifier.

Algorithm 2: `Interjection_word_start` (IWS)

Input: Tweets that start with interjection words.
Output: Classification of tweets as sarcastic or not sarcastic.
Notation: ADJ: adjective, V: verb, ADV: adverb, N: noun, UH: interjection, T: tweets, C: corpus, TF: POS Tag file, t: tag, TWT: tweet wise tag, FT: first tag, INT: immediate next tag, NT: next tag.
Initialisation : TF = { ϕ }

1: **for** T in C **do**
2: k = `find_postag` (T)
3: $TF \leftarrow TF \cup k$
4: **end for**
5: **for** TWT in TF **do**
6: t = `find_subset` (TWT)
7: FT = `find_first_tag` (t)
8: INT = `find_immediate_next_tag` (t)
9: NT = `find_next_tag` (t)
10: **if** $FT = UH$ && $INT = ADJ \parallel ADV$ **then**
11: Tweet is sarcastic
12: **else if** $(FT = UH)$ && $(NT = (ADV + ADJ) \parallel (ADJ + N) \parallel (ADV + V))$ **then**
13: Tweet is sarcastic
14: **else if** $(FT \neq UH)$ **then**
15: Invalid tweet.
16: **else**
17: Tweet is not sarcastic
18: **end if**
19: **end for**

Algorithm 2 (IWS) says, find the POS tag of given input corpus in step 2 and collect it in the tag file. Step 6 finds a tweet wise subset of tag from tag file and extracts very first tag of every tweet in step 7. Similarly, step 8 finds an immediate next tag for every tweet. If first tag is an interjection (UH) and an immediate next tag is either adjective or adverb than tweet is classified as sarcastic. Otherwise, find further next tag (other than first tag) of every tweet in step 9. If first tag is an interjection (UH) and next tag is either (ADV + ADJ) or (ADJ + N) or (ADV +V) than tweet classify as sarcastic. If first tag is not an interjection than invalid input otherwise tweet is not sarcastic.

V. Results and Discussion

This section describes the results of the proposed algorithm. We started with database collection followed by testing the performance of both proposed algorithms. Finally, compared our result with the state-of-art.

A. Database Collection

For training, 50000 tweets are collected with the sarcasm hashtag (#sarcasm) from Twitter with keyword love, amazing, good, hate, sad, happy, bad, hurt, awesome, excited, nice, great, sick, etc. For testing, Tweets are collected in two categories (i) tweets with sarcasm hashtag and (ii) tweets without hashtag. To test Algorithm 1 (PBLGA), 1500 random tweets collected with sarcasm hashtag and 1500 random tweets collected without hashtag are used. Similarly, to test Algorithm 2 (IWS), 1000 tweets were collected with sarcasm hashtag and 2500 tweets without any hashtag. For Algorithm 2, both 1000 and 2500 tweets start with an interjection word. The collected tweets were preprocessed to remove hashtags, URL, @user, upper case word to lower case from the tweet corpus.

B. Results

Algorithm 1 was used on a corpus of 50000 collected tweets to generate the lexicon of positive sentiment, negative situation, negative sentiment, positive situation. Few sample sentiment word and situation phrase are shown in TABLE VI.

The observations of PBLGA are as follows:

- 1064 tweets identified as a sarcastic out of 1500 tweets with hashtag sarcastic and ground truth is 1200.
- 420 tweets identified as sarcastic out of another 1500 tweets without sarcastic hashtag and ground truth is 360.
- Total 1484 tweets identified as sarcastic out of 3000 tweets and ground truth was 1560.

About 3500 interjection tweets are tested with Algorithm 2. The observations are:

- 826 tweets identified as sarcastic out of 1000 tweets start that with interjection word and sarcasm hashtag and ground truth is 733.
- 999 tweets were identified as sarcastic out of 2500 tweets starting with interjection word and no sarcastic hashtag and ground truth is 1045.

- Total 1825 tweets identified as sarcastic out of 3500 tweets start with interjection word.

TABLE VI: Sample learned phrases by PBLGA

Positive sentiment phrase: Great, lucky, cute, good, nice, happy, glad, delicious, 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, etc.
Negative sentiment phrase: 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: no 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 released, 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.

TABLE VI. shows sample learned phrases by PBLGA in four categories, namely (1) Positive sentiment phrase from the positive sentiment file (PSF), which sentiment values are greater than 0. (2) Negative sentiment phrase from the negative sentiment file (NSF), which sentiment scores are less than 0. (3) Positive situation phase from positive situation file (psf), which sentiment scores are greater than 0. (4) Negative situation phase from negative situation file (nsf), which having sentiment scores less than 0.

C. Comparison of results with state-of-art

To compare the results with existing work, three parameters are considered, namely, $precision$, $recall$ and $f-score$. The formula to calculate $precision$, $recall$ and $f-score$ shown in Equations 4, 5 ,6 respectively.

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

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

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

where,
T_p = True positive, F_p = False positive, F_n = False negative

$Precision$ 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 (T_p) as algorithm and ground truth, both identified positive and the tweets are incorrectly identified called false positive (F_p) as algorithm identified positive but ground truth was negative. Similarly, how many of tweets correctly rejected is called true negative (T_n) as both algorithm and ground truth identified negative and how many tweets incorrectly rejected called false negative (F_n) as algorithm identified negative but ground truth was positive.

TABLE VII: Confusion matrix of tweets in different categories

Category of tweets	T_p	F_p	F_n	T_n
1500 tweets with sarcasm hashtag	955	109	218	218
1500 tweets without sarcasm hashtag	270	150	90	990
1000 interjection tweets with sarcasm hashtag	706	120	26	146
2500 interjection tweets without sarcasm hashtag	772	227	272	1182

Confusion matrix for tweets in four categories, namely 1500 tweets with sarcasm hashtag, 1500 tweets without sarcasm hashtag, 1000 interjection tweets with sarcasm hashtag and 2500 interjection tweets without sarcasm hashtag. Tweets of first and second category tested through PBLGA, 1064 & 420 tweets are identified as sarcastic respectively. Similarly, Tweets of third and fourth category tested through IWS, 826 & 999 tweets are identified as sarcastic respectively. We used three annotator to find the ground truth of tweets in above categories. According to annotator, ground truth is 1200, 360, 733 and 1136 for first, second, third and fourth category respectively. Based on annotator's ground truth, calculated true positive, true negative, false positive and false negative for all four categories are shown in TABLE VII.

TABLE VIII: $Precision$, $Recall$ and $F-score$ of tweets in different categories

Category of tweets	$Precision$	$Recall$	$F-score$
1500 tweets with sarcasm hashtag	0.89	0.81	0.84
1500 tweets without sarcasm hashtag	0.64	0.75	0.69
1000 interjection tweets with sarcasm hashtag	0.85	0.96	0.90
2500 interjection tweets without sarcasm hashtag	0.77	0.73	0.74

TABLE VIII shows the $precision$, $recall$ and $f-score$ for tweets in all above four categories. These statistical values calculated using Equations 4, 5 and 6.

A comparison of the proposed algorithms (PBLGA and IWS) with existing state-of-art algorithm is shown in TABLE IX. The comparison was done with various statistical parameters such as $precision$, $recall$, and $f-score$ and observed that

TABLE IX: Comparison of proposed methods with state-of-art

$Approach$	$Precision$	$Recall$	$F - score$
Barbieri *et al.* system	0.88	0.87	0.88
Tungthamthiti *et al.* system	0.76	0.76	0.76
Riloff *et al.* system with positive verb	0.28	0.45	0.35
with negative situation	0.29	0.38	0.33
Contrast (+VPs, -situation)unordered	0.11	0.56	0.18
Contrast (+VPs, -situation)ordered	0.09	0.70	0.15
Contrast (+preds, -situation)	0.13	0.63	0.22
Liebrecht *et al.* system with 50/50	0.75	-	-
with 25/75 neg, pos ratio	0.56	-	-
PBLGA with sar tweets	**0.89**	0.81	0.84
PBLGA without sar tweets	0.64	0.75	0.69
IWS sarcastic tweets	0.85	**0.96**	**0.90**
IWS without sarcastic tweets	0.77	0.73	0.74

PBLGA achieved $precision$ = 0.89 with tweets has sarcastic hashtag. IWS achieved $recall$= 0.96 and $f - score$= 0.90 with the sarcastic hashtag tweet.

VI. Conclusion

Sarcasm analysis is one of most important and challenging task as it doesnt have any pre-defined structure. None of the researchers got the complete solution in this field. Researchers continuously trying to get better results. In this paper, we proposed two algorithms to identify sarcasm for two different types of tweet structure. Firstly, tweet structure having contradiction between negative sentiment and positive situation. Secondly, Tweet starts with interjection word. We focused on tweets which starts with interjection word. Algorithm 1 applied in the first type of tweet structure and Algorithm 2 applied in second types of tweet. Algorithm 1 achieved 0.89, 0.81 and 0.84 $precision$, $recall$ and $f - score$ respectively in tweets with sarcastic hashtag and 0.64, 0.75 and 0.69 $precision$, $recall$ and $f - score$ respectively in tweets without sarcastic hashtag. Algorithm 2 achieved 0.85, 0.96 and 0.90 $precision$, $recall$ and $f - score$ respectively in tweets with sarcastic hashtag and 0.77, 0.73 and 0.74 $precision$, $recall$ and $f - score$ respectively in tweets without sarcastic hashtag. Testing result of algorithm 2 proves Lunandos statement as they said " if the text is using interjection words, the text has more tendency to be classified into sarcastic. Our second algorithm identified 999 sarcastic tweets out of 2500 random tweets without any hashtag, while 826 identified sarcastic out of 1000 tweets with sarcastic hashtag. Sarcasm detection in audio clip, images are still open area. In the future, we will work on this area with some regional language.

References

[1] R. González-Ibánez, S. Muresan, and N. Wacholder, "Identifying sarcasm in twitter: a closer look," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-vol 2*, pp. 581–586, 2011.

[2] R. J. Kreuz and G. M. Caucci, "Lexical influences on the perception of sarcasm," in *Proceedings of the Workshop on computational approaches to Figurative Language, ACL*, pp. 1–4, 2007.

[3] C. Liebrecht, F. Kunneman, and A. van den Bosch, "The perfect solution for detecting sarcasm in tweets# not," *Association for Computational Linguistics*, pp. 29–37, 2013.

[4] 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 *EMNLP*, pp. 704–714, 2013.

[5] E. Lunando and A. Purwarianti, "Indonesian social media sentiment analysis with sarcasm detection," in *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on IEEE*, pp. 195–198, 2013.

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

[7] D. Davidov, O. Tsur, and A. Rappoport, "Semi-supervised recognition of sarcastic sentences in twitter and amazon," in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pp. 107–116, 2010, Association for Computational Linguistics.

[8] E. Filatova, "Irony and sarcasm: Corpus generation and analysis using crowdsourcing.," in *LREC*, pp. 392–398, 2012.

[9] R. Justo, T. Corcoran, S. M. Lukin, M. Walker, and M. I. Torres, "Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web," *Knowledge-Based Systems*, vol. 69, pp. 124–133, 2014.

[10] F. Kunneman, C. Liebrecht, M. van Mulken, and A. van den Bosch, "Signaling sarcasm: From hyperbole to hashtag," *Information Processing & Management (in press)*, 2014.

[11] P. Liu, W. Chen, G. Ou, T. Wang, D. Yang, and K. Lei, "Sarcasm detection in social media based on imbalanced classification," in *Web-Age Information Management*, pp. 459–471, 2014.

[12] D. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis," in *Proceedings of LREC*, pp. 4238–4243, 2014.

[13] O. Tsur, D. Davidov, and A. Rappoport, "Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews.," in *ICWSM*, pp. 162–169, 2010.

[14] D. Tayal, S. Yadav, K. Gupta, B. Rajput, and K. Kumari, "Polarity detection of sarcastic political tweets," in *Computing for Sustainable Global Development (INDIACom), 2014 International Conference on IEEE*, pp. 625–628, 2014.

[15] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm detection on twitter: A behavioral modeling approach," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pp. 97–106, 2015.

[16] P. Tungthamthiti, K. Shirai, and M. Mohd, "Recognition of sarcasm in tweets based on concept level sentiment analysis and supervised learning approaches," pp. 404–413, 2014.

[17] R. J. Kreuz and R. M. Roberts, "Two cues for verbal irony: Hyperbole and the ironic tone of voice," *Metaphor and symbol*, vol. 10, no. 1, pp. 21–31, 1995.

[18] A. Utsumi, "Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony," *Journal of Pragmatics*, vol. 32, no. 12, pp. 1777–1806, 2000.

[19] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: Liwc 2001," *Mahway: Lawrence Erlbaum Associates*, vol. 71, pp. 1–11, 2001.

[20] C. Strapparava, A. Valitutti, *et al.*, "Wordnet affect: an affective extension of wordnet.," in *LREC*, vol. 4, pp. 1083–1086, 2004.

[21] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.