


Sarcasm Analysis and Mood Retention Using NLP Techniques


Srijita Majumdar, St. Xavier's College (Autonomous), Kolkata, India

Debabrata Datta, St. Xavier's College (Autonomous), Kolkata, India

Arpan Deyasi, RCC Institute of Information Technology, India

 <https://orcid.org/0000-0003-3056-896X>

Soumen Mukherjee, RCC Institute of Information Technology, India

 <https://orcid.org/0000-0002-8211-0507>

Arup Kumar Bhattacharjee, RCC Institute of Information Technology, India

Anal Acharya, St. Xavier's College (Autonomous), Kolkata, India

ABSTRACT

Sarcasm detection in written texts is the Achilles' heel of research areas in sentiment analysis, especially with the absence of the rightful verbal tone, facial expression, or body gesture that leads to random misinterpretations. It is crucial in sectors of social media, advertisements, and user feedback on services that require proper interpretation for service evaluation and improvisation of their products. The objective here thereby is to identify sarcasm within a given text by experimenting with the original predicted mood of the text and work on its transformation with the several variations in combination of the standard sarcastic elements present in the corresponding writing. Here standard NLP techniques are used for identification and interpretation. This involves detecting primary connotation of the given text (e.g., positive/neutral/negative), followed by detecting elements of sarcasm. Then, under the presence of the sarcasm indicator algorithm, the rightful interpretation of the previously detected mood is attempted.

KEYWORDS

Connotation, Context of Usage, Emoticons, Final Sentiment, P-O-S Tagging, Sarcasm, Sarcasm Detector, Sarcastic Elements, Sentiment Polarity, Sentiment Transformation, Stemming

INTRODUCTION

With the advent of technology, social networking websites have become a popular medium for the users to express their sentiments, thoughts and opinions on various topics of interest which may either be any event or be any person or product. Social media channels have also got a foothold as far as the discussions amongst people on various trendy topics are concerned. People throughout the world

DOI: 10.4018/IJIRR.289952

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

can freely express their ideas with this type of platform to convey their outlook on any topic on earth. This has also become a platform for interacting with others. With this, the volume of data generated has become enormous. Users post more than 340 million tweets and 1.6 billion search queries every day (Bharti, Vachha, Pradhan, Babu & Jena, 2016). With this huge amount of data being generated, a number of challenges crop up. These challenges include accessing, storing and processing of data, verification of data sources, dealing with misinformation. There is another challenge in dealing with unstructured data as on an average, almost 80% of generated data are of this form.

In social media platforms, user interactions may vary from a simple text messaging to interacting through different multimedia channels such as images and videos. Users are often engaged in providing real-time feedback about any current and happening event all around the globe. In the e-commerce platforms, users often play a vital role in submitting feedbacks which help companies and franchises with invaluable data that can be used to manipulate their products as per the market scenario accordingly.

At the same time, it is quite obvious that an opinion or a review posted through any social networking site may not always directly state the exact orientation of the users but it may sarcastically express the true feelings. Sarcasm is a kind of sentiment which acts as an interfering factor in any text that can flip the polarity. If in a tweet or post, a word expresses a positive sentiment in a negative context, the tweet or the post should be classified as a sarcastic one. Unlike a simple negation, sarcastic tweets contain positive words. It is also true that an exaggerated variation of a positive word may convey a negative opinion or vice versa. So there is a requirement of analyzing this large volume of text in the form of reviews, tweets or feedback messages to gauge their precise orientation. To achieve this, each text may have to pass through a set of relevant algorithms.

However, finding and verifying the authenticity of opinions or reviews is not an easy task. Also, it is hard to comprehensively analyze all the reviews and recognize sarcasm in them. This is specifically true in case of product reviews. Any error in analysis may end up misleading them. The usage of feedback information is necessary in data-driven sectors in correspondence to Artificial Intelligence, which leads to several security and privacy concerns. Block-chain Techniques used for implementing several security methods ensures the provision of the rightful interpretations of information to customers from companies and vice versa (Cuong, Kumar & Solanki, 2020).

The term ‘sarcasm’ is generally used to mock, often with satirical or ironic remarks, with a purpose to amuse and hurt someone, or some section of society, simultaneously. This can also reflect a state of ambivalence and can be expressed variably through speech and text. Within straight talks, sarcasm is deluded by facial expressions and body gestures while within audio or telephonic conversations, sarcasm can be inferred by alterations in tone of voice. Thus sarcasm is one of the challenging areas faced in the field of sentiment analysis especially in text. It is the most difficult to identify compared to other methods, mainly due to absence of the verbal tone, thereby causing a frequent misinterpretation in conversations mostly done via writing i.e. Letters, emails, social media posts etc. (Literary Devices, Definition and Examples of Literary Terms, n.d.).

However, there are certain standards that have been set by which sarcasm can be identified and can be conveyed. For example, by using uppercase words or letters, excessive usage of exclamatory marks that might indicate certain exaggeration in the tone of the message, usage of emoticons etc. (Peng C., Lakis M., Pan J., 2015). The application areas of sarcastic text detection are also significant. It may be used for letting the reviewer know the intent of the writer and the context in which it is said. The main objective of our present work is to identify the presence of sarcasm in a given text.

The work mentioned here in contrast to the major number works falling under the related field, develops certain basic contexts of presence of Sarcasm and uses the presence of sarcastic elements to make a text fit into that context.

The Sarcastic elements constitute a greater portion of a pre-ordained database that is formed with user-intervention thereby allowing all the basic forms of interpretations in various possible contexts.

User opinion in decision-making during categorization of sarcastic elements was probed to be necessary since the interpretation of a word varies dynamically with situations, incidents and various other contexts, that vary dynamically with changing times and circumstances. Also, the algorithm `Sarcasm_detector()` preserves stop words to a certain extent so that they may later assist an adjoining connotation to the rightful interpretation of sarcasm as will be demonstrated hereafter.

BACKGROUND

- **Basic Approach:** A sarcasm detection algorithm should always be accompanied by an effective sentiment polarity prediction algorithm/procedure that would interpret the rightful sentiment of the text that otherwise seems to convey a different sentiment, after certain elements of sarcasm have been detected in the text. Every work in this area of research corresponds to this idea. Also, it should be accompanied by a formidable database of classified resources of text and parts-of-speech and other methodologies that would enable the proper classification between sarcastic and non-sarcastic texts.
- **Illustrated works:** One such approach has been taken up by Mäntylä et al. (2018), is developing where a language dependent model has been developed, using an assortment of different language resources like lexicon of markers, sentiment lexicon, morphological dictionaries and WordNet based ontology, to classify text into the sarcastic and the non-sarcastic kind. Also, these approaches use various features like antonymous pairs i.e. the pair of words with opposite polarities, POS tagging and irony markers (Mäntylä, Graziotin & Kuuttila, 2018).

Again, in the published work of Bharti *et al.* (2016), a Hadoop based outline is proposed that captures real time tweets and processes it with a set of identification algorithms. The Hadoop based framework proves to be much more dynamic and outperforms other conventional methods for capturing and analyzing real-time tweets (Bharti et al., 2016).

In the research work of Gupta and his co-workers (2019), the Artificial Neural Network is used to classify polarities to words in their algorithm for sarcasm detection (Gupta, Singh & Singla, 2019). In analyzing emoticons for sentiment analysis and sarcasm detection, Rodrigues *et al.* (2018) presented a set of emojis along with different norms which are used already by a section of users. The data have been gathered from LEED (Lisbon emojis and emoticon) that includes 85 number of emoticons, 238 stimuli, and 153 numbers of emojis (Rodrigues, Prada, Garrido & Lopes, 2018).

SHC (Sarcasm hash tag classifier) works on the hash-tag sentiment analysis in order to make classification between various types of sarcasm and non-sarcasm tweets. The function of SHC depends upon the SHI (Sarcasm hash tag indicator) and the relation aiming the hash tag and tweets. Mhatre et al., 2017, presented a pre-processing scheme which is applied before testing of the text (Mhatre, Phodnekar, Kadam, Chawathe & Ghag, 2017).

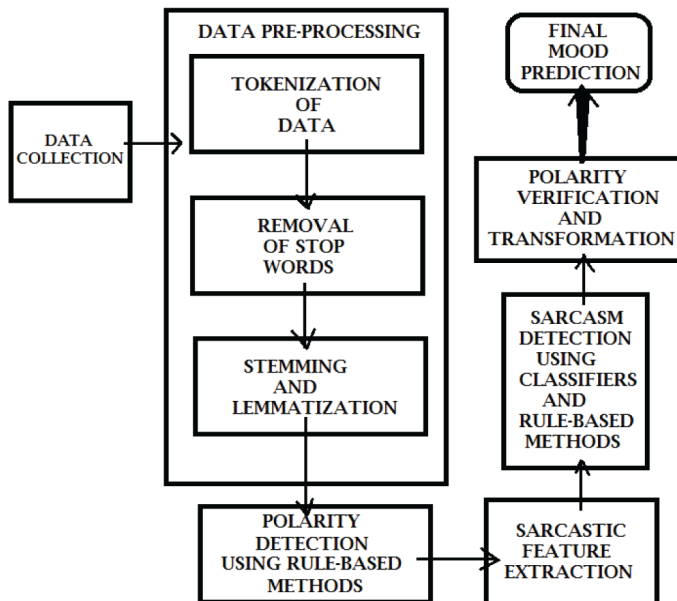
PROPOSED METHODOLOGY FOR SARCASM DETECTION

The procedure that has been proposed in the current research work hereby goes by the following stages as shown in Figure 1.

Stage Descriptions

1. **Data Collection:** Random user inputs need to be collected in order to achieve variety and enable the algorithm to gain an all-round perspective about the language in general.
2. **Data Pre-Processing:** This stage can be sub-divided into four different sub-catagories based on the task/job-role namely tokenization, filtering, lemmatization and stemming:

Figure 1. Schematic flow of sarcasm detection for sentiment analysis



- a. 'Tokenization' is the job where any sequence or character is broken up into string of pieces, called tokens. This will further be processed.
- b. 'Stop Words' are those meaningless stemmed words, words with neutral connotations, and it effectively reduces the processing time with the numbers which are filtered away. It also helps to analyze the respective placements of different connotations and sarcastic elements (Datta, Majumdar, Sen & Sen, 2019, p. 1336).
- c. 'Stemming' method has the sole objective to obtain the stem (root) of derived words (Datta et. al., 2019, p. 1336-1337).
- d. 'Lemmatization' is the special type of job which taken into account the morphological analysis of the words. This can be expressed in elaborative manner. In this task process, different inflected forms of a word are grouped together so that it can be analyzed as a single item. This will eventually leads to many-to-singular form mapping (Datta et. al., 2019, p. 1337).
3. **Polarity Detection Using Rule-Based Methods:** A suitable sentiment analysis algorithm is used to classify the tokens into their suitable category and analyze their placements in order to have the rightful sentiment polarity of the text (Datta et. al., 2019, p. 1340-1341).
4. **Sarcastic Feature Extraction and Detection:** Certain standard pre-conceived elements of sarcasm has to be recognized within the given text with the help of a pre-trained categorized database and their relations with the sentiment of the text extracted above has to be analyzed along with a rule based technique that has been developed using a thorough analysis of English text and the general interpretation of words in various contexts, to give the desired results and sentiment polarity of the sarcastic text.
5. **Polarity Verification and Transformation:** A final round of sentiment polarity verification is done in order to reflect back the changes in sentiment polarity that might have occurred with the detection of the sarcastic elements. If no elements get detected then the original sentiment polarity is assigned to the text. The sentiment analysis algorithm is used again to generate a single sentiment polarity for the entire input by analyzing the mutual placements of the individual sentences of the text.

Elements of Sarcasm to be Noted Within a Given Text

1. **Modern/Millennial Abbreviations:** The abbreviations used now-a-days can be used as a considerable measure to estimate the presence of a sarcastic element within a text given the presence of the general notion of their context of usage, which is mostly bent towards an exaggeration of the mood of the writer or an implicit sarcasm. However, based on their context of usage, there are mostly two categories of millennial abbreviations:
 - a. **Positive abbreviations:** They generally represent a positive sentiment:
Ex – BAE, TTYL, LIT, OOTD etc. (Nasir, K.G. and Al-Ghizzy, M.J.D., 2019; Wallace, 2016)
 - b. **Negative or Sarcastic Abbreviations:** While some abbreviations from this category are generally meant to represent a negative sentiment, others are popularly acclaimed to be of the sarcastic genre. Since, sarcasm mostly evolves within a negative context of usage, their implications are almost indistinguishable, when it comes to interpretation of the sentiment that they portray:

Ex – LOL, LMAO, ROFL etc. (Nasir, K.G. and Al-Ghizzy, M.J.D., 2019; Wallace, 2016)

2. **Uppercase Words (Excluding Abbreviations And ‘I’ And ‘A’):** Usually such form of representation of words is meant to imply an exaggeration of sentiment or the opposite notion of the general connotation of the word mentioned:

Ex – STATEMENT: Yes, she is a GOOD teacher! (*Implying that she is the complete opposite of ‘good’*)

3. **Emoticons and Their Context Of Usage:** Emoticons are shorthand forms of facial expressions such as:-) which provides an allowance for expressing one’s feelings, moods and emotions by providing a non-verbal context to a written message. It helps catch the reader’s attention and enhances one’s understanding of the sentiment of the written text which it accompanies. It facilitates more expressive messages. An emoticon is a graphic symbol that represents not only facial expressions, but also concepts and ideas, such as celebration, weather, vehicles and buildings, food and drink, animals and plants, or emotions, feelings, and activities (Novak, Smailović, Sluban & Mozetič, 2017).

There are basically three categories of emoticons, based on their generic usage:

1. Positive emoticons:

Ex –:D,;),:-), <3 etc.

Figure 2. Elements of sarcasm

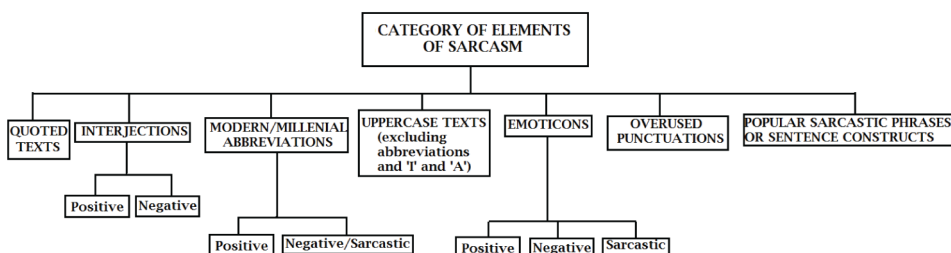


Table 1. General association of sentiment polarities with emoticons (Sindhu, Vadivu & Rao, 2018, p. 435)

	Emoticons	Associated sentiment words from the random input samples
Positive	:) :D =)	happy, good, fantastic, wonderful, lovely, amazing, awesome etc.
	:) :-) :-) :-D =D ; P =] 8) (:	smile, friends, favorite, music, heart, kind, positive, weekend, spring etc.
Negative	: (: / XD : ' : ' (: - (D: ; (: - / : l : \	sorry, sad, miss, hate, shit, ugly, broke, late, sick etc.
Sarcastic	: P ; D : - P :] : p	can't, LOL, never, feel, look, what etc.
	XP	shoot, stuck, fatally, helpless etc.
	DX	music, smartphone, camera etc.

2. Negative emoticons:

Ex – :(, :-), :', - (etc.

3. Sarcastic emoticons:

Ex – :P, XD, -_-, :| etc.

While there are certain clichéd emoticons that are associated with sarcastic statements providing an instant recognition, others usually are tagged as sarcastic in context of their usage.

The three basic ways in which emoticons imply sarcasm are:

- Classes of emoticons are inherently associated with sarcasm (e.g. XD, ;), :P etc.).
- Creating a negative and a positive pool of emoticons with random user inputs. If a statement of a negative sentiment is accompanied by a positive emoticon or a statement of a positive sentiment is accompanied a negative emoticon, then it may imply sarcasm:

Ex – “I couldn’t be happier working in the IT firm! Such a productive life, I have! ☺”

- Also, if an emoticon (+ve, -ve or otherwise) is accompanied by UPPERCASE/italicized/double quoted words/statements or the aforementioned abbreviations (e.g. LOL etc.) in a statement, then its usage may be classified as sarcastic:

Ex-”I’m glad we’re having a rehearsal dinner. I rarely practice my meals before I eat. ;)”

4. **Overused Cases of Punctuation Marks:** When a sentence is accompanied by the presence of multiple punctuation marks instead of the grammatically legitimate single punctuation, one might interpret it to be a sarcastic statement (Peng C. et al.,2015):

Ex – “Really Sherlock?? No, I agree, you ARE clever!!”

Here, what apparently might strike as grammatical errors, are taken as elements of a sarcastic statement with heavy punctuations.

5. **Usage and Context of Interjections:** A statement can be implied as sarcastic when it is of a negative sentiment and is accompanied by a positive interjection and vice-versa:

Ex – “Yay! Yay! What could be better than the sole of your favourite sports-shoe, coming off right in the middle of the marathon?!”

6. **Popular and Widely Used Sarcastic Phrases or Sentence Constructions:** There are certain stereotypical expressions or statements that are often widely used or are associated with sarcasm. An inherent pool of all these expressions may lead to a quicker detection of sarcasm:

Ex – The phrase ‘could it be <comparative statement>’ e.g.

- “Could it be any simpler?”
- “Could it be more beautiful?” or Statements prefixed by ‘As if/Like’ e.g.
- “As if you desire to do all the cooking by yourself!”
- “Like she wants to tolerate the chaotic students in the class” etc. or “The day she gets a job is the day when pigs will fly” etc.

7. **Quoted Words or Sentences:** Quoted words within a sentence always are meant to assert a certain higher level of implication upon it thereby often seemingly contradicting their general sensation:

Ex – I “accidentally” happened to stumble upon your chocolate cake in the refrigerator and hence decided to try it (Implying the opposite sense i.e. the work had been done very much by intention and not by accident).

While Interjections, Punctuation Marks, Quotes, Intensity words constitute the *Hyperbolic elements*, emoticons form the *Pragmatic elements* of Sarcasm Detection (Bharti et al., 2016).

The other categories like *Lexical elements* that includes Uni-gram, Bi-gram and Tri-gram, that shall remain undemonstrated owing to their absence in usage for the algorithm mentioned here (Bharti et al., 2016).

DETECTING SARCASM

Predicting the Mood of the Given Input Text From Its Connotation

1. **Connotation Prediction:** Analysis of a line text has to be done at first, for predicting just the general connotation/sentiment polarity (positive, neutral, negative) of the text, with a suitable sentiment analysis algorithm. An algorithm called *Mood_Sense()* is used for serving the purpose (Datta et al., 2019, p. 1340-1341).
2. **Detection and Interpretation of Mood From the Placement/Presence of the Sarcastic Elements:** A sentence denoting a certain kind of connotation, neutral or otherwise, derived from Datta et al. (Datta et al.2019), Table 1, if accompanied by any one of the sarcastic elements e.g. a sarcastic millennial abbreviation, a negative emoticons, a sarcastic emoticons, an uppercase word, quoted word or a typical sarcastic sentence construct, has its connotation and emotional quotient reverted back to the appropriate sort of polarity of interpretation in accordance to the sarcasm implied.

Thus, the resultant connotation obtained from the second last column in the above table may be interpreted as the appropriate interpretation of the rightful sentiment polarity (+ve/-ve/neutral)

of the corresponding input statement in the last column, with respect to the presence of the rightful combination of sarcastic elements in it.

Algorithm For Sarcasm Detection

Algorithm Sarcasm_Detector(Input_Text)

Begin

```
//Let n sentences be considered to be constituted in the input text.  
//List A[n] is assumed to contain all the n sentences of the Input separately.  
//List Final[n] is to be considered to contain all the final  
connotations of the corresponding statements in //the list A[n]  
after the evaluation of sarcasm.
```

```
//Algorithm Mood_Sense() has been used to obtain the connotation and the  
final mood of a given statement in A[i] by deciding upon the placement of  
connotations and other parts of a sentence (Datta et al., 2019)
```

```
    for (i=1 to n)  
        Original_Connotation=Mood_Sense(A[i]);  
        If (Sarcastic_Element(A[i])==1)  
//to detect the presence of any of the aforementioned elements of sarcasm  
            Final[i]=Alter_Connotation(Table 2, A[i], Original_Connotation)  
        Else  
            Final[i]=Original_Connotation  
    End of For-Loop  
    Final_Sentiment= Mood_Sense(Final[])
```

End

Description of Sarcasm_Detector()

- **Input:** The input to the above algorithm is presumed to be given in a normal sentence construct, where the sentence count is kept as n, e.g. an input with 3 sentences will have n=3. A sentence is considered to have been completed when a single or a bunch of generic *sentence terminators* such as- *Note of Interrogation(?)*, *Full Stop (.)*, *Note of Exclamation(!)* etc. - is encountered.
- **Data Structure:** A[n] is a list of size 'n' is thus created where the ith sentence is stored at the ith location, to be analyzed individually for the presence of sarcasm. Also, Final[n] is a list that stores the individual connotation of the sentences.
- **Procedure:** A looping construct is used to traverse every sentence of the input one-by-one, where algorithm *Mood_Sense()* is used to predict the 'Original Connotation' i.e. connotation sans the elements of sarcasm, of the sentences (Datta et al., 2019):
 - Method *Sarcastic_Element()* is used to detect the presence of any elements of sarcasm in the ith sentence and return an indicator accordingly.
 - Method *Alter_Connotation()* takes the ith sentence and it's corresponding 'Original Connotation' as input and uses the aforementioned *Table 2*, to perform the transformation of connotation, if the presence of one or more sarcastic element is found within a given text. In the absence of a sarcastic element, the algorithm assigns the sentence to its pre-affirmed 'Original Connotation', as was predicted by 'Mood_Sense()'.
 - *Final_Sentiment* is the variable that stores the Final prediction when all the 'n' connotations of all the 'n' sentences are analysed adjacently to give the final connotation of the entire input text using the generic placement pattern given in *Table 1* in (Datta et al., 2019, p. 1340, Step 5.3.).

The sole reason that the working principle of *Mood_Sense()* remains unaffected by sarcasm is because it's designed to dismiss the sarcastic elements as neutral connotations that do not in any manner add on to the interpretation of its sentiment. Also, it's database remains devoid of the rightful categorizations for the sarcastic elements as listed above, in Section 3.1. Therefore, the predictability of *Mood_Sense()* is accomplice by the further specification of Sarcasm_Detector() to enhance its precision of interpretation by performing the transformation of the '*Original Connotation*' if need arises and a sarcastic element gets detected. In the presence of a sarcastic element the '*Original Connotation*' gets reversed to its polar opposite as is illustrated in table 2 as shown in the next page.

Table 2. Interpretation of mood in accordance to the placement of sarcastic elements

	Original connotation (Without the sarcasm indicator algorithm) of the input text	Coexistence of either of the mentioned elements of sarcasm in the corresponding input text	Connotation predicted as a result of both the sentiment analysis and the sarcasm indicator algorithm	Illustrations
CASE I	N_a or N_b or N_c	Sarcastic Emoticons	P_b	STATEMENT: Yeah, like she could 'fail' in an exam she has been working so hard upon for such a long time? :)
		Quoted Text or extra punctuations		
CASE II	P_b or Neutral	Negative/Sarcastic Millennial Abbreviations	N_c	STATEMENT: Need money for college. Need college for a job. Need a job for money. Seriously, who WAS the mastermind behind this system???:'(
		Uppercase or Italicized Text		
		Negative Emoticons		
		Sarcastic Emoticons		
		Quoted Text or extra punctuations		
CASE III	Neutral	Negative/Sarcastic Millennial Abbreviations	N_c	STATEMENT: My 'attitude' in exams. They give me questions I don't know. I give them answers they don't know. LOL!
		Uppercase or Italicized Text		
		Sarcastic Emoticons		
		Quoted Text or extra punctuations		
		Negative Interjections		
CASE IV	Neutral	Positive Millennial Abbreviations	P_b	STATEMENT: We write to taste life twice, in the moment and in retrospect:D
		Positive Emoticons		
		Positive Interjections		
		Quoted Text or extra punctuations		
CASE V	P_a	Negative/Sarcastic Millennial Abbreviation	N_a	STATEMENT: I am free of all prejudice. I hate every idiot equally. FML! :-/
		Upper Case Text		
		Negative Emoticons		
		Sarcastic Emoticons		
		Negative Interjections		
		Popular Sarcastic Sentence Constructs		

Illustration of the Procedure of Mood Retention Using 'Sarcasm_Detector()' Algorithm

INPUT TEXT: *"No problem! I NEVER feel insecure at being ignored!:P !"*

SET $n=3$, A[3], Final[3] for the three separate sentences in the given input text.

SENTENCE: A[1]

A[1]=*"No Problem!"*

No- Na
Problem-Nb

Mood_Sense(Na+Nb)=Pa
Therefore, Original_Connotation=Pa... (A)

(Sarcastic_Element(A[1])!=1) //no sarcastic element detected

Therefore, Final[1]=Original_Connotation=Pa

SENTENCE: A[2]

A[2]=*" I NEVER feel insecure at being ignored!"*

I- Neutral
NEVER- Na
feel- Neutral
insecure- Na
at- Neutral
being- Neutral
ignored- Nb

Mood_Sense(Neutral+Na+Neutral+Na+Neutral+Neutral+Nb)=Pa
Therefore, Original_Connotation=Pa... (B)

Sarcastic_Element(A[2])==1) //unabbreviated Upper Case text is detected as the sarcastic element

Therefore, Final[2]= Alter_Connotation(Table 2, A[2], Pa)

Alter_Connotaion= Table 2, CASE V

Therefore, Final[2]= Na

SENTENCE: A[3]

A[3]=*":P!"*

:P- Neutral

Mood_Sense(Neutral)=Neutral

Therefore, Original_Connotation=Neutral... (C)

(Sarcastic_Element(A[3])==1)//Sarcastic Emoticon detected as the
sarcastic element

Therefore, Final[3]= Alter_Connotation(Table 2, A[3], Neutral)

Alter_Connotaion= Table 2, CASE III

Therefore, Final[2]= Nc

Thus, the list Final[] ends up looking like:

P_a	N_a	N_c
1	2	3

Thus, by further using the Algorithm Mood_Sense() and referring to the placement of connotations from Table 1, it is derived that the connotation of the entire given text, after the Sarcasm Analysis is completed and the contents of the Final[] list is obtained.

Connotation of the entire text, with reference to Table 1, obtained is:

$(P_a + N_a) + N_c = N_b + N_c = N_c$, i.e., Negative Sentiment, (Datta et al., 2019, p. 1340, Step 5.3.)

and by the presence of certain Sarcastic Elements detected within the writing, it can be rightfully concluded that a rather negative sentiment has been expressed with Sarcasm.

Note: On, the contrary, had the original connotation of the statements been taken into consideration, before implementing Algorithm Sarcasm_Detector(), from (A), (B) and (C), then the total sentiment of the entire text that would have been predicted, with reference to Table I, would be:

$P_a + P_b + \text{Neutral} = P_b + \text{Neutral} = P_b$, i.e., Positive Sentiment

which going by the context of the given input text, would have been incorrectly deciphered.

The presence of any other *Intensity words* in the given text can be further inspected, with reference to a pre-trained database, and the algorithm Mood_Sense() can be used once again, to rightfully, come up with the perfect result for the mood of the given text.

Table 3. Demonstrating the difference in the output obtained before and after the application of Sarcasm_Detector() upon the aforementioned piece of text

OUTPUT (Without Sarcasm_Detector())	OUTPUT (With Sarcasm_Detector())
Connotation-Positive	Sarcasm Indicated. Connotation- Negative.
Mood- Happy	Mood- Critical or Angry or Confused
Vulnerability Level- None	Vulnerability Level- Mild

RESULTS OBTAINED AND FURTHER ANALYSIS

Training

Basically, to train a data in this context would refer to catching any hint of sarcasm in the given input text, which can only be done efficiently if there is a pre-ordained database present with a significant amount of sarcastic elements like emoticons, phrases, millennial abbreviations etc. categorized accordingly along with the words in the input text being put into their rightful connotation categories by the *Mood_Sense()* algorithm.

If the connotation implied by the *Mood_Sense()* algorithm are valid then the mood prediction can be done most effectively. Thereafter, if the sarcastic elements get detected rightfully in the given input, then the rightful combination of the original connotation and the combination of the Sarcastic Elements would give us the outcome determining the presence of sarcasm. Thus, with a combinatory effect of grammatical, etymological rules and some amount of a generic user perception, a certain level of effectiveness has been achieved. Therefore, after extracting each word and its stem form, the following algorithm has been incorporated.

A word can simultaneously be categorized into its rightful connotation and also the rightful sarcastic element.

Ex – ‘Shit!’ can fall in both the categories of N_c (3rd Category negative) and the millennial slang, at the same time (Datta et al., 2019).

Algorithm for Training and Categorization of Words With User Intervention

CATEGORIZE_USER_INPUT ()

Algorithm CATEGORIZE_USER_INPUT ()

Begin

Step 1. Initialize categorized = 0.

Step 2. STEM is inquired to be categorized as -

-A positive or negative connotation.

-A positive or negative interjection.

-A sarcastic element

Step 2.1. If (User_Decide (STEM) = True)

Step 2.1.1. STEM is placed in one or more category.

Step 2.1.2. Set categorized = 1.

Step 2.1.3. STEM is checked for belonging to any of the categories of 'Intensity Words'.

Exit. //if stem falls into anyone or more category

Step 2.2. Else //if the stem obtained remains uncategorized

Step 2.2.1. WORD is enquired to be categorized as-

-A positive or negative connotation

-A positive or negative interjection.

-A sarcastic element

Step 2.2.1.1. If (User_Decide (WORD) = True)

Step 2.2.1.1.1. WORD is placed in one or more category.

Step 2.2.1.1.2. Set categorised = 1.

Step 2.2.1.1.3. WORD is checked for belonging to any of the categories of 'Intensity Words'.

Exit. //if word falls into anyone or more category

Step 2.2.1.2. Else

Step 2.2.1.2.1. WORD is placed in neutral category.

Exit if-else.

End.

Brief Mentions From CATEGORIZE_USER_INPUT ():

- **Categorised:** A flag variable that indicates that a word has been made to fall within a particular category by the user himself.
- **User_Decide ():** A method that accepts a word or its stem and returns ‘True’ when the user comes upon with the category and sets the newly found word to it.
- The *Porter Stemmer Algorithm* has been used for obtaining the required stem of the words in a given sentence. A word is extracted at first and is stemmed down with the Porter Stemmer Algorithm. If the stem is detected to be falling under any of the categories of connotations/intensity words/neutral words etc., then the *categorized* flag variable for that corresponding word having being set, the control proceeds on to the next word, without checking for the category of the previous, original word anymore. Only if the *stem* derived from the original word remains uncategorized, the original *word* is checked to be placed under the aforementioned categories of types of words.
- A word, derived from a given sentence, can fall under or more category. Accurate categorization of words is important for the algorithms- Mood_Sense() (Datta et al., 2019, p. 1340) and Sarcasm_Detector() to obtain the rightful combination and placement of word types, in order to obtain the original sentiment polarity and thereafter detect the presence of sarcasm and perform the rightful interpretation of the *Mood* of the statements, accordingly.

Effects of Training on a Sample Data

Demonstrated below is a set of statements given as input to algorithm combination MOOD_SENSE () and Sarcasm_Detector() before and after the training process.

Testing

Testing of the algorithm can be done by giving some random English statements as inputs and checking the results, and the accuracy. When the model is passed the validation process, and the database has been updated, it is ready for accepting data.

Table 4. Before training

No.	Input Statement	Elements of Sarcasm Detected	Final Connotation	Mood Determination	Vulnerability level	Validity of results
1.	Yeah, like I WOULD apologize for something I didn't do -_- !	✓	-ve	Annoyed/ Confused	Mild	✓
2.	When people ask me stupid questions, it is my legal obligation to give a sarcastic remark.	×	-ve	Stressed	Mild	✓
3.	It's okay if you don't like me. Not everyone has 'good taste'. LOL!	✓	-ve	Melancholic	Intermediate	✓
4.	Mirrors can't talk, lucky for you they can't laugh either! JEEZ!	×	+ve	Happy	None	×
5.	Everyone seems normal until you get to know them.	×	Neutral	Static	None	×

Table 5. After training

No.	Input Statement	Elements of Sarcasm Detected	Final Connotation	Mood Determination	Vulnerability level	Validity Of results
1.	Yeah, like I WOULD apologize for something I didn't do -- !	✓	-ve	Annoyed/ Confused	Mild	✓
2.	When people ask me stupid questions, it is my legal obligation to give a sarcastic remark.	×	-ve	Stressed	Mild	✓
3.	It's okay if you don't like me. Not everyone has 'good taste'. LOL!	✓	-ve	Melancholic	Intermediate	✓
4.	Mirrors can't talk, lucky for you they can't laugh either! JEEZ!	✓	-ve	Stressed	Mild	✓
5.	Everyone seems normal until you get to know them.	×	Neutral	Static	None	×

The proposed algorithm was implemented using OOP with jdk8.0. The obtained results are shown in the Table 6.

COMPARING THE PROPOSED METHOD WITH AN EXISTING WORK

The work *Sarcasm Detection in Twitter using Sentiment Analysis* by Bala Durga et al. (June, 2019) (Dharmavarapu & Bayana, 2019) uses the Naive Bayes' classifier and the Adaboost ML algorithm to classify sarcastic elements and non-sarcastic elements.

The greatest disadvantage being that it cannot incorporate feature interactions and may utilize binning as a tool to assign discrete classes to the sarcastic elements thereby throwing away a lot of important information. It merely does this by taking count of elements from various classes, calculating their probabilities and then sticking to the class that shows the greatest probability to identify the sentiment of a given sentence. This approach even though may work fine in purely descriptive sentences, it may fail to give a correct outcome when used in wider range of context and in sentences that portray a greater emotional variety, as the interpretation of words vary not only with their frequency but also mostly with their context of usage.

The output classifies the sarcastic and non-sarcastic tweets from the given list of tweets. Sentiment analysis organizes the tweets into positive tweet or negative tweet. Naive Bayes' classification algorithm finds the sarcastic tweets by considering the probabilities.

The occurrence of a particular tweet, falling under one of the five categories of topics is demonstrated in the form of approximated percentages in the table.

In contrast, the work mentioned hereby varies in two basic respects and mainly in classifying text into its most basic interpretations with respect to the following aspects:

- **Mood Retention with User Intervention:** The Sentiment Analysis algorithm *Mood_Sense()* aids in the process by classifying some basic context of mood retention and enabling the most

Table 6. Results obtained after testing

No.	Input Statement	Elements of Sarcasm Detected	Final Connotation	Mood Determination	Vulnerability level	Validity of results
1.	A room without books is like a body without a soul.	×	P _b	Satisfied	None	✓
2.	Silence is golden. Duct tape is silver!!!	✓	N _c	Stressed	Mild	✓
3.	Anyone who thinks sitting in church can make you a Christian must also think that sitting in a garage can make you a CAR! ;)	✓	N _c	Stressed	Mild	✓
4.	Insanity is doing the same thing, over and over again, but expecting different results.	×	N _a	Angry	Critical	✓
5.	You know you're in love when you can't fall asleep because reality is finally better than your dreams.	×	P _b	Cheerful	None	✓
6.	Don't feel bad, I'm usually about to die:P	×	P _b	Exuberant	None	×
7.	Like I'd want to go out with a guy who doesn't take any pleasure in a good book?! GOSH! He must be so intolerably ignorant!!	✓	N _a	Extremely Shocked or Angry	Extreme	×
8.	I am busy right now, can I ignore you some other time?	×	N _a	Stressed	Mild	✓
9.	Could I be in love with deadlines any more? I just love the whooshing noise they make as they go by.	✓	N _c	Worried	Mild	✓
10.	Violence won't solve anything. But it sure makes me feel good. LMAO!	✓	P _b	Exuberant	None	×

justified transformation. The interpretation of sentences and their connotations are extremely diverse and vary exceedingly with the context. Therefore, a minimum level of user-intervention is necessary to aid in the classification of words with respect to their generic context.

Thus, after each word and consequently each statement has been allotted to its rightful connotation (i.e. +ve, neutral or -ve), *Mood_Sense()* classifies the context of usage into some of the most generic

Table 7. Tabular representation of the work carried at (Dharmavarapu & Bayana, 2019)

TWEET CATEGORY	TOPIC #0 %	TOPIC #1 %	TOPIC #2 %	TOPIC #3 %	TOPIC #4 %	TOPIC #5 %
tw 40978	10	60	10	10	10	60
tw 13274	50	10	10	50	50	10
tw 30644	10	40	40	10	60	10
tw 30641	20	96	10	10	10	10
tw 21545	8	56	60	40	8	40
tw 32589	30	30	30	30	30	80
tw 26430	20	60	16	16	16	50
tw 0564	20	50	20	20	20	40
tw 33140	30	30	54	30	30	30
tw 12371	64	10	80	10	70	14

mood interpretations (e.g. Happy, Annoyed, Satisfied, Depressed, Cheerful etc.) and allocates the input to the category of mood that matches the correct intensity by using elements of hyperbole such as interjections, comparatives, superlatives etc.

Also, at a given point of time, a word can be categorized under two or more classes of connotation, and the rightful classification is used for varying contexts during sentiment analysis. Much importance is given to the context of interpretation by allowing a justified amount of user intervention and thus the predictability of the *Mood_Sense()* algorithm increases with the increase in the number of attempts in testing and training.

- **Context of sarcasm and its interpretation:** *Sarcasm_Detector()* uses the approach where it works with some basic contexts of interpretation of sarcasm with respect to the presence of some basic sarcastic elements and develops categories with respect to all their possible combinations.

The presence of a particular combination indicates the presence of sarcasm in the statement.

Also, while the former algorithms the work by Bala Durga et al. (2019) uses a POS tagger to classify stop words from the connotative ones, *Mood_Sense()* and *Sarcasm_Detector()* conjures up their interpretation under a pre-trained and categorized database, that stores words into their rightful categories as per their generic interpretation with user intervention, without taking into consideration the part-of-speech it belongs to.

- **Presence of stop words:** The stop words are given a significant notice in case of Sarcasm detection rather than being thrown entirely out of consideration. Rather they are assigned neutral connotation so that whenever they fall into the same context with yet another connotative word it can both associate in the interpretation of the connotation as well as the nature of sarcasm.

Illustrating case 3 from Table 2, it is observed that a statement that was initially interpreted to be of a neutral category or a statement completely devoid of connotative words, when associated with the corresponding combination of the sarcastic elements gets interpreted into one of the negative connotations, generically observing as shown in Table 8. Thereby this illustrates the importance of the preservation and the justified analysis of stop words.

Table 8. Change of interpretation

	Original connotation	Combination of elements of sarcasm	Final interpretation	Demo
CASE III	Neutral	Negative/Sarcastic millennial abbreviations	N _c	STATEMENT: My 'attitude' in exams. They give me questions I don't know. I give them answers they don't know. LOL!
		Uppercase or italicized text		
		Sarcastic emoticons		
		Quoted text or extra punctuations		
		Negative interjections		

Both the algorithms- *Mood_Sense()* and *Sarcasm_Detector()* work in close proximity utilizing a Hyperbolic and Pragmatic elements of Sarcastic text for analysis, the results of which are demonstrated here above in Section 5.1.2.

The work by Jha et. al., (2018), 'A Novel Approach on Visual Question Answering by Parameter Prediction using Faster Region Based Convolutional Neural Network' basically deals with the process with which a machine can explore several aspects of an image by referring to suitable natural language questions and their corresponding answers. This process, combines aspects of both NLP and Computer Vision (CV), is usually referred to as a VQA (Visual Questioning Answering) that utilizes Convolution Neural Network (CNN) for visual representation and Recurrent Neural Network (RNN) with Long Short Term Memory (LSTM) for textual (question) representation. Dataset is prepared of Triplets (Question, answer and image).

The important aspect being interpreting and gathering data from the questions and putting the derived data to the most suitable future interpretation, which will form the basis of differentiation of interpretation of text. The main probe that will be performed here is into the RNN, LSTM and Word Embedding methodologies of deriving several aspects of a written text and making long-term decisions thereafter.

Transformation Method 1: RNN (Recurrent Neural Network)

It is used for learning the sequence of data like series of frames, video, text etc. Since, random statements are taken as input here, can be broken down into tokens that will be useful for deriving implications that vary over time. Conceptually it differs from the standard neural network as the inputs are usually done in single words rather than in sentences but then wing to its flexibility in wording with varying lengths of sentences it provides an additional advantage of sharing features attributed across various variety of texts.

Transformation Method 2: LSTM (Long Short Term Memory)

It can process both single data points (e.g. a word or an image) and entire sequences of data (such as a speech or a video).

A generic LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. They are well-suited to classifying, processing and predicting results based on time series data since there can be lags of unknown duration between important events in a time series.

A cell-state is a long-term memory, it represents all the learning at over time, and hidden state (is like a current memory). It is used to categorize between relevant part of a network-which one to retain and which one to forget.

The basic procedures before feeding into the LSTM include:

- **Tokenization:** The questions are tokenized into sentences by splitting the words with space and punctuation and any number considered as words. Extra words are discarded.
- **Word embedding:** Word embedding is then applied on each word, and the results are reflected into their corresponding vectors, before being fed into the LSTM.

Certain implementations of the LSTM, as have been mentioned in Table 1, Jha et. al, 2018, has a knowledge base that signifies facts about the world. It has an inference engine for reasoning about the facts by using rules and logic and deduces new facts.

Transformation Method 3: Word Embedding

Word embedding- Words are represented as a vector of real numbers. Words, representing related concepts belong to the same vector, by projecting a word from a different dimension equivalent to the vocabulary length to a lower dimensional space. The size of the vector varies from the vocabulary size used in various systems.

LIMITATIONS AND FUTURE SCOPE

The above analogy looks for several elementary constituents to indicate the presence of sarcasm. However, in the real scenario of the language implementation and understanding, a sentence may also be sarcastic in the total absence of these components.

Ex – *STATEMENT: The day she will get a proper job is the day when sandcastles will remain standing forever.*

This statement, despite being completely devoid of every aforementioned element still implies the sense of sarcasm, in the effect that ‘sandcastles’ are related to a totally irrelevant physical phenomenon i.e. ‘standing forever’.

Also, at times the coexistence of a positive and a negative connotation may imply sarcasm.

Ex – *I couldn't be happier to be burdened to this extent!*

However, statements like- *Even though her marks were bad, she didn't lose hope!* go against setting a proper convention for the above.

To come into a definite conclusion is quite complex as it requires further analysis of random inputs from the real life implementation of sarcasm in sentences and speeches to justify the context in which a word is used. POS tagging, knowledge representation and mapping, and several other supervised and unsupervised machine learning algorithms may be made useful in this regard.

Also, despite both the algorithms– *Mood_Sense()* and *Sarcasm_Detector()* demonstrating the classification of connotations and elements of sarcasm with the help of user intervention, the method becomes susceptible to certain erroneous interpretations as well, which would result in a consequential erroneous detection. So, there remains a tradeoff between achieving the rightful context of interpretation and the time consumed by the gradual training process.

The Importance and Usage of P-O-S Tagging in Sarcasm Detection (Sindhu et al., 2018)

The working principle of the algorithm *Sarcasm_Detector()* does not use a POS tagger here, however, often incongruity in statements forms the primary source of sarcasm that can be analyzed with proper POS tagging done by a suitable POS tagger, and therefore it could have been made useful in several respects.

Table 9. Comparison of the CNN methodologies with the proposed methodology

Text Interpretation Methodologies of Neural Network			Proposed Methodology for Sarcasm Detection
1.	RNN	<p>1. This architecture is capable of capturing the dependencies, but only in one direction of language. Basically, solely in case of Natural Language Processing, it is assumed that the later word has no effect on the meaning of the priorly stated word, which by the human interpretation of language text, isn't justified. Bidirectional RNN is thus separately suggested for this purpose.</p> <p>2. RNN isn't god at capturing long-term dependencies.</p>	<p>1. Here, the methodology has conjured up all the possible occurrences of words belonging to various connotations (From table 3 in Datta et. al, 2019) and sarcastic forms, and have generalised patterns in which they may be identified in text, thereby giving it a two-way approach of interpretation. If a sentence is priorly detected to be of a perfectly normal interpretation devoid off sarcasm, the detection of any sarcastic parameter by the Algorithm Sarcasm_Detector(), at the start/within/end of the statement, thereafter, changes the entire interpretation of the text to it's polar opposite connotation or otherwise as given in table 2.</p> <p>2. Works with a given piece of ext at a given point of time. The subsequent additions are further appended thereafter and the entire text is reevaluated from the point where it had been left off.</p>
2.	LSTM	<p>1. LSTMs require lots of resources and time to get ready for real world applications. They require high-memory bandwidth that the system usually fails to provide for.</p>	<p>1. With the rise of data mining, developers look for a model that is not system-dependent and can retain past evaluations for a longer time than LSTM. The proposed methodology manifests this by dividing the sentences into smaller parts (e.g. connotations, interjections, superlatives, sarcastic elements etc.) for easy remembrance and retain past interpretations from tables that demonstrate the basic varieties of patterns of occurrences of elements.</p>
3.	Word Embedding	<p>1. In reality these dimensions are not that clear or easily understandable. This does not conjure a problem as the algorithms train on the mathematical relationships between the dimensions. What is represented by the dimension is meaningless for a neural network from training and prediction point of view</p>	<p>1. The proposed methodology is independent of such procedures of mapping closely related vocabulary into vectors of real-numbers.</p>

Note: Since the basic objective of the final works of both the papers are not very closely related, the comparisons performed here are of the effectiveness of the methodologies used rather than of the final results.

Incongruity basically concerns a sudden change in sentiment polarity from one sentence to its subsequent statement or within a line of text itself. It usually happens in between the same parts-of-speeches e.g. two nouns, two verbs, two adjectives etc.

Types of Incongruities

- **Explicit Incongruities:** Overtly expressed through the sentiment of words of opposite polarities. Often constitutes two nouns, verbs, adjectives etc. of opposite polarity placed within a single statement:

Ex – “I love being ignored!” where “love” is a verb +ve polarity and “ignore” is a verb of negative polarity.

- **Implicit Incongruities:** Covertly expressed through phrases of implied sentiment:

Ex – “He saw a bright future ahead, for himself and decided to reach there faster by consuming cyanide”

While explicit incongruities maybe applied in the most basic stages using a suitable NLP algorithm, implicit incongruities require a thorough study of the history of the person/object, to catch any hints of sarcasm.

As mentioned and explained in (Esuli & Sebastiani, 2006), classes 1 to 7 can be classified under explicit incongruities and can be satisfactorily implied using NLP algorithms. Some of these classes e.g. classes 2 and 3 as stated in (Esuli & Sebastiani, 2006) contain inter-sentential incongruity. Algorithms, concerning these classes may work by eliminating stop-words, punctuations and extract phrases, analyze their sentimental polarity and mutual placements and categorize them into one of the listed classes. Class 8 falls under the implicit incongruous class of statements, needs a gradual understanding, learning of facts and historical records concerning the corresponding person/object, spoken about in the statement.

CONCLUSION

The algorithm here deals with the contexts of interpretation of sarcasm with respect to the presence of some basic sarcastic elements by develops categories with respect to all their possible combinations. These categories are developed with the aid of some thorough analysis of texts across a number of domains and genres. The database developed and categorized gradually, with respect to the several variety of connotations, types of intensifiers and sarcastic elements, yields a better precision of outcome with the increase in the number of datasets (which is approximately 1500 variety of articles and writings) and by employing a suitable algorithm (Section 5.1.1) that allows newfound words to be put into their rightful dataset categories. Also, one word may exist in one or more category with its variety of interpretations in various contexts. This approach contributes towards the development in the transition types in Table 2, with which an input text can be analyzed from several perspectives, irrespective of the occurrence of the sarcastic elements. In the end, the placement table (Datta et al., 2019, Table 1, p. 1339) allows for the proper interpretation of the context. The entire approach here focuses more on the context of an element more than its frequency of occurrence.

Sarcasm detection and analysis in social media provides invaluable insight into the current public opinion on trends and events in real time. In this paper, an investigation has been made on some of the characteristics of sarcasm on social networking platforms and described some preliminary experiments which study the effect of sarcasm on sentiment analysis. In particular, the present research work is concerned not just with identifying whether the texts are sarcastic or not, but also considering the range of the sarcastic modifier on the meaning of it and on the polarity of the sentiment expressed. The initial observations are that there are many interesting phenomena to be observed, and that detection of sarcasm, while useful, is not sufficient for accurate sentiment analysis and adding further rules to deal with the scope of sarcastic texts does, however, improve performance considerably.

REFERENCES

- Bharti, S. K., Vachha, B., Pradhan, R. K., Babu, K. S., & Jena, S. K. (2016). *Sarcastic Sentiment*. <https://www.sciencedirect.com/science/article/pii/S235286481630027X>
- Bhattacharyya, P. (2017). *Sarcasm Detection: A Computational and Cognitive Study*. Department of Computer Science and Engineering, IIT, Madras. <https://nlp.stanford.edu/seminar/details/pbhattacharyya.pdf>
- Cuong, N. H. H., Kumar, G., & Solanki, V. K. (2021). Blockchain-Based Digital Rights Management Techniques. In *Large-Scale Data Streaming, Processing, and Blockchain Security* (pp. 168–180). IGI Global. Retrieved from <https://www.igi-global.com/chapter/blockchain-based-digital-rights-management-techniques/259470>
- Datta, D., Majumdar, S., Sen., O., & Sen, A. (2019). Mood and Vulnerability Prediction through Natural Language Processing. *International Journal of Innovative Technology and Exploring Engineering*, 9(2). <https://www.ijitee.org/wp-content/uploads/papers/v9i2/B6158129219.pdf>
- Dharmavarapu, B. D., & Bayana, J. (2019). Sarcasm Detection in Twitter using Sentiment Analysis. *International Journal of Recent Technology and Engineering*, 8(4), 642-644. <https://www.ijrte.org/wp-content/uploads/papers/v8i4s4/A11180681S419.pdf>
- Esuli, A., & Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. *Proceedings of Language Resources and Evaluation Conference*, 417 – 422. <http://www.lrec-conf.org/proceedings/lrec2006/pdf/384.pdf>
- González-Ibáñez, R., Muresan, S., & Wacholder, N. (2011). Identifying sarcasm in twitter: a closer look. *HLT '11: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 581 – 586. <https://www.aclweb.org/anthology/P11-2102.pdf>
- Gupta, A. (2018). *Sarcasm Detection: Achilles Heel of sentiment analysis by at #ODSC_India* [Video file]. https://www.youtube.com/watch?v=FoR_-ELAcTE&t=1s
- Gupta, S., Singh, R., & Singla, V. (2020). Emoticon and Text Sarcasm Detection in Sentiment Analysis. In A. Luhach, J. Kosa, R. Poonia, X. Z. Gao, & D. Singh (Eds.), *First International Conference on Sustainable Technologies for Computational Intelligence. Advances in Intelligent Systems and Computing* (Vol. 1045, pp. 1–10). Springer. Retrieved from https://link.springer.com/chapter/10.1007/978-981-15-0029-9_1
- Ide, N., Macleod, C., & Grishman, R. (2001). The American national corpus: a standardized resource of American English. *Proceedings of Corpus Linguistics*. <http://www.lrec-conf.org/proceedings/lrec2000/pdf/196.pdf>
- Jha, S., Dey, A., Kumar, R., & Solanki, V. K. (2018). A Novel Approach on Visual Question Answering by Parameter Prediction using Faster Region Base Convolutional Neural Network. *International Journal of Interactive Multimedia and Artificial Intelligence*, 5(5). Retrieved from- <https://www.ijimai.org/journal/bibcite/reference/2688>
- Literary Devices. (n.d.). *Definition and Examples of Literary Terms*. <https://www.literarydevices.net/sarcasm/>
- Mäntylä, M. V., Graziotin, D., & Kuuttila, M. (2018, February). Using Lexical Resources for Irony and Sarcasm Classification. *Computer Science Review*, 27, 16–32. <https://dl.acm.org/doi/pdf/10.1145/3136273.3136298>
- Maynard, D., & Greenwood, M. A. (2014). *Who cares about Sarcastic tweets? Investigating the impact of sarcasm on Sentiment Analysis*. University of Sheffield, Department of Computer Science, IREC. <https://gate.ac.uk/sale/lrec2014/arcomem/sarcasm.pdf>
- Mhatre, M., Phondekar, D., Kadam, P., Chawathe, A., & Ghag, K. (2017). Dimensionality reduction for sentiment analysis using pre-processing techniques. *Proceeding of IEEE International Conference on Computing Methodologies and Communication (ICCMC)*. <https://ieeexplore.ieee.org/document/8282676>
- Murthy, D., Allu, S., Andhavarapu, B., & Bagadi, M. (2020). Text based Sentiment Analysis using LSTM. *International Journal of Engineering Research*. https://www.researchgate.net/publication/341873850_Text_based_Sentiment_Analysis_using_LSTM
- Nasir, K.G., & Al-Ghizzy, M.J.D. (2019). Linguistic Features of Netspeak: Abbreviations, Acronyms and Punctuation Marks. *University of Thi-Qar Journal of education for Humanities Science*, 1(1), 1-69.

Nasukawa, T., & Yi, J. (2003). Sentiment Analysis: Capturing Favourability Using Natural Language Processing. *Proceedings of the 2nd International Conference on Knowledge Capture*, 70-77. <https://dl.acm.org/doi/10.1145/945645.945658>

Nigam, V. (2019). *Natural Language Processing: From Basics to Using RNN and LSTM*. <https://towardsdatascience.com/natural-language-processing-from-basics-to-using-rnn-and-lstm-ef6779e4ae66>

Novak, P. K., Smailović, J., Sluban, B., & Mozetič, I. (2015, December). Sentiment of Emojis. *PLoS One*, 10(12), e0144296.

Pal, S., Ghosh, S., & Nag, A. (2018). Sentiment Analysis in the Light of LSTM Recurrent Neural Networks. *International Journal of Synthetic Emotions*, 9, 33–39. doi:10.4018/IJSE.2018010103

Peng, C., Lakis, M., & Pan, J. (2015). *Detecting Sarcasm in Text: An Obvious Solution to a Trivial Problem*. Stanford CS 229 Machine Learning Final Project. Retrieved from- http://cs229.stanford.edu/proj2015/044_report.pdf

Rodrigues, D., Prada, M., Gaspar, R., Garrido, M. V., & Lopes, D. (2018). Lisbon emoji and emoticon database (LEED): Norms for emoji and emoticons in seven evaluative dimensions. *Behavior Research Methods*, 50(1), 392–405. <https://link.springer.com/article/10.3758/s13428-017-0878-6>

Sindhu, C., Vadivu, G., & Rao, M. V. (2018). A Comprehensive Study on Sarcasm Detection Techniques in Sentiment Analysis. *International Journal of Pure and Applied Mathematics*, 118, 433 – 442. <https://acadpubl.eu/hub/2018-118-22/articles/22a/63.pdf>

Solanki, V. K., Cuong, N. H. H., & Lu, Z. J. (2019). Opinion Mining: Using Machine Learning Techniques. In *Extracting Knowledge From Opinion Mining* (pp. 66–82). IGI Global. Retrieved from <https://pure.hud.ac.uk/en/publications/opinion-mining-using-machine-learning-techniques>

Wallace, B. (2016). *Do You Speak Millennial? A Modern Acronym Guide*. <https://medium.com/digital-vault/do-you-speak-millennial-a-modern-acronym-guide-da45c367b8e3>

Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *WIREs Data Mining Knowl Discov.*, 8, e1253. 10.1002/widm.1253

Debabrata Datta pursued Master of Technology from University of Calcutta, India and he is currently pursuing his Ph.D. in Technology from the same university. He is an Assistant Professor in the department of Computer Science, St. Xavier's College (Autonomous), Kolkata, India. He is a life member of IETE. He has published more than twenty-five research papers in different reputed international journals and conferences mainly in the field of data analysis which is his primary area of research interest. He has more than twelve years of teaching experience both in the undergraduate and post graduate level of Computer Science. He has more than seven years of research experience.

Arpan Deyasi is presently working as Assistant Professor in the Department of Electronics and Communication Engineering in RCC Institute of Information Technology, Kolkata, INDIA. He has 13 years of professional experience in academics and industry. He received B.Sc (Hons), B.Tech, M.Tech Degree from University of Calcutta. His work spans around in the field of semiconductor nanostructure, semiconductor photonics. He has published more than 150 peer-reviewed research papers. His major teaching subjects are Solid State Device, Electromagnetics, Photonics. He is associated with different International and National Conferences in various aspects, and organized several technical events under the banner of IE(I) Kolkata section. He is reviewer of a few journals of repute and some prestigious conferences in INDIA and abroad. He is senior member of IEEE, secretary of IEEE Electron Device Society (Kolkata Chapter), member of IE(I), Optical Society of India, IETE, ISTE etc. He is working as SPOC of RCCIIT Local Chapter (NPTEL course), Nodal Coordinator of Internshala, e-outreach programme (IITB) and IISR (under ISRO), Admin of Coursera campus program and Faculty Adviser of the student chapter of Institution of Engineers (INDIA) in ECE Department at RCCIIT.

Soumen Mukherjee did his B.Sc (Physics Honours) from Calcutta University, M.C.A. from Kalyani University and ME in Information Technology from West Bengal University of Technology. He is the silver medalist in ME examination from the university. He has done his Post-Graduate Diploma in Business Management from Institute of Management Technology, Center of Distance Learning, Ghaziabad. He is now working as an Assistant Professor in RCC Institute of Information Technology, Kolkata. He has more than 17 years teaching experience in the field of Computer Science and Application. He has over 40 research paper published in different National and International Journal and Conferences and Book Chapter. He has total 9 chapters in different books under International Publisher. He has contributed in over 20 internationally acclaimed books in the field of Computer Science and Engineering. He has got two best paper awards in the conference and seminar. He has also edited 3 books. His research fields are Image Processing, Machine Learning and Pedagogy. He is a life member of several institutions like IETE, CSI, ISTE, FOSET, etc.

Arup Kumar Bhattacharjee did his graduation from University of Calcutta, Master of Computer Application from University of Kalyani and M. Tech. from West Bengal University of Technology. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering, RCC Institute of Information Technology, Kolkata, India. He is teaching core and electives courses in undergraduate and postgraduate programs in the field of Computer Science for last 18 years. He has published many research papers in different National and International Journals and Conferences. He has edited 2 books and contributed in over 20 internationally acclaimed books in the field of Computer Science and Engineering. His studies continue in the areas of soft and evolutionary computing, object-oriented technology.

Anal Acharya completed his MCA from Jadavpur University, Kolkata, India in 1997 and subsequently joined St. Xavier's College, Kolkata as lecturer in Computer Science. He completed his ME in IT stream from West Bengal University of Technology, Kolkata in 2006. He was the Head of the Department of Computer Science from the period of 01.01.2009 to 30.06.2013. He was awarded Ph.D. (Tech.) in Computer Science and Engineering in 2017 for his thesis titled "A Study on Data Mining Techniques in Web based Learning". Continuing in this vein his present research interest is Educational and Social Data Mining. He has above 23 years of experience in undergraduate and post graduate teaching & supervised several post graduate dissertations. Currently he is designated as an Assistant Professor in Department of Computer science in St. Xavier's College, Kolkata. He has more than 50 papers in International Conferences, Journals and Book Chapters.