Context-based Sarcasm Detection in Hindi Tweets

1st Santosh Kumar Bharti
Dept. of CSE
NIT Rourkela
Rourkela, India
sbharti1984@gmail.com

1st Korra Sathya Babu
Dept. of CSE
NIT Rourkela
Rourkela, India
prof.ksb@gmail.com

2nd Rahul Raman
Dept. of CSE
VIT Vellore
Vellore, India
rahulraman2@gmail.com

***Abstract*—Sentiment analysis is the way of finding ones' opinion towards any specific target. Sarcasm is a special type of sentiment which infers the opposite meaning of what people convey in the text. It is often expressed using positive or intensified positive words. Nowadays, posting sarcastic messages on social media like Twitter, Facebook, WhatsApp, *etc.*, became a new trend to avoid direct negativity. In the presence of sarcasm, sentiment analysis on these social media texts became the most challenging task. Therefore, an automated system is required for sarcasm detector in textual data. Many researchers have proposed several sarcasm detection techniques to identify sarcastic text. These techniques are designed to detect sarcasm on the text scripted in English since it is the most popular language in social networking groups. However, parallel research for sarcasm detection on different Asian languages like Hindi, Telugu, Tamil, Urdu, and Bengali are not yet explored. One of the reasons for the less exploration of these languages for sarcastic sentiment analysis is the lack of annotated corpus even though they are popular in a large networked society. In this article, we proposed a context-based pattern *i.e.* "sarcasm as a contradiction between a tweet and the context of its related news" for sarcasm detection in Hindi tweets. The proposed approach utilized Hindi news as the context of a tweet with in the same timestamp and attained an accuracy of 87%.**

***Index Terms*—Hindi, Online News, Sarcastic, Sentiment, Social Media, Tweets**

I. Introduction

Online companion has gained tremendous momentum in recent times for business, politics, entertainment, *etc.* Social media such as Twitter, Facebook, WhatsApp, *etc.*, is considered as the popular medium for online companion and it attain the response of users from worldwide. These responses include ones' sentiment or opinion towards any specific target such as individuals, events, topics, products, organizations, services, *etc* [1]. The sentiment is nothing but an opinion of any individual towards a specific target. It may be either positive, negative or neutral. Manual extraction of the sentiment of the social media text is a tedious job for individuals as well as organizations. There is a need for an automated system which will be capable of providing sentiment of social media text without any human interference.

Sentiment analysis is a part of Natural Language Processing (NLP) that deals in finding the orientation of an opinion in a piece of text about any topic [2]. The presence of sarcastic text in the corpus makes sentiment analysis challenging as most of the existing sentiment analysis system does not consider sarcastic sentiments. Due to this, most of the existing systems for sentiment analysis fail in detecting the sarcastic sentiment. Sarcasm is a special kind of sentiment that usually flips the orientation of the opinion in a given piece of text. People often express sarcasm verbally through the use of heavy tonal stress and certain gestural clues like rolling of the eyes, hand's movement, *etc.* These tonal and gestural clues are obviously missing while expressing sarcasm in text, which makes its detection even more difficult task. Sometimes, human beings feel difficulty to understand sarcasm in text. The sarcastic sentence usually looks positive, but overall meaning becomes negative due to the presence of sarcasm. An automated system is required for sentiment analysis which will be capable of identifying sarcastic sentiment as well.

In recent past, many researchers have focused on sarcasm detection and proposed automatic sarcasm detector in text [3]–[12]. These systems are developed for sarcasm detection in text scripted in English. English is the most popular language across the world, and in this domain, plenty of resources are freely available for research. Therefore, majority of researchers have preferred English domain for their research. Many other languages are getting popular in rapid pace such as Hindi, Arabic, Dutch, Mandarin, etc. These languages fall in low resourced categories as the availability of free resources in this domains are very rare.

In low resourced languages, Hindi is the fourth-most spoken language in the world, after Mandarin, Spanish and English [13]. It has 490 million speakers across the world, and majority of them are from India [14]. It is widely used for speaking in countries like India, Mauritius, Fiji, Suriname, Guyana, Trinidad & Tobago and Nepal [15]. These days' in India, Hindi is getting more popularity on social media such as Facebook, Twitter, WhatsApp, etc. People are posting messages, comments very frequently in the Hindi language. With the increased amount of information being communicated via regional languages like Hindi on social media, there comes a promising opportunity of mining this information. In order to mine the Hindi information automatically from social media, various NLP tasks such as part-of-speech (POS) tagging, sentiment analysis had been already developed.

For sarcasm detection in Hindi, a system was developed [16] for Hindi tweets using a similar set of features used for English tweets namely, #tag, emoticons, punctuation marks

etc. However, in real scenario natural Hindi tweets [1] are different in structure unlike English scripted tweets or Hindi tweets translated from English tweets. A sample list of Hindi sarcastic tweets is shown in Fig. 1. To identify sarcasm in such natural Hindi tweets, the same feature set used for English scripted tweets might not be applied effectively. Therefore, one needs to rely on other parameters such as news context, specific patterns, rules, *etc.*, to identify sarcastic Hindi tweets.

1. काले धन पे पेनल्टी 200% से घटा के 10% कर दी? काला धन वालों के सामने मोदी जी ने घुटने टेक दिए?- @ArvindKejriwal
2. दो दिन बाद शाहरुख खान अपना 51वां जन्मदिन मनाने वाले हैं, लेकिन उनकी हीरोइन की उम्र लगातार कम होती जा रही है
3. @Rajrrsingh #सुना_है! #iphone7 टिम कुक के टकले पे रख के चार्ज किया जायेगा!
4. आज सुबह मुझे सवच्छता भारत अभियान सड़क पर बिखरा हुआ मिला! #swachbharat #Hindi #clean #mock #sarcasm
5. #JioOffer का आधा से ज्यादा डेटा तो लोग सिर्फ ट्विटर पे अरविन्द केजरीवाल को ट्रोल करने में इस्तेमाल करते है.

Fig. 1. A sample Hindi sarcastic tweets.

In this article, we proposed a context-based pattern "sarcasm as a contradiction between a tweet and the context of its related news" for sarcasm detection in Hindi tweets. The proposed approach utilized Hindi news as the context of a tweet with in the same timestamp [2]. In this approach, we collected one liner news from Twitter Hindi news sources such as ABP News हिंदी @ABPNewsHindi, आज तक @aajtak, *etc.* Similarly, a corpus of Hindi tweets is collected based on collected news related keywords in the same timestamp. Next, we identify the sentiment of a tweet and the context of related news. If both contradict, then the tweet is classified as sarcastic. Otherwise, the tweet is non-sarcastic.

The rest of the article is organized as follows: Section II describes related work. Preliminaries are explained in Section III. The proposed scheme is discussed in Section IV. Analysis of the results are given in Section V and the conclusion of the article is drawn in Section VI.

II. Related Work

In recent times, several studies about sarcasm detection in text scripted in the English language have explored as English is a resource rich language [3]–[11]. The domain of low resourced languages such as Hindi, Tamil, Telugu, Urdu, Arabic, Indonesian, Mandarin, *etc.*, is yet to be explored. Some of the previous authors explored these languages for sarcasm detection [16]–[19] where availability of datasets are very rare.

A sarcasm detector was proposed [17] for Indonesian social media to identify sarcasm in Indonesian tweets using interjection words such as 'aha', 'wow', 'nah', etc. They have collected only 980 and 300 Indonesian tweets manually from Twitter for training and testing respectively. They concluded that, if a tweet contains an interjection word then there is high tendency to be sarcastic. Similarly, a multi-strategy ensemble classification algorithm was proposed [18] for Chinese social media to identify sarcasm in Chinese tweets using a comprehensive sarcasm feature set including lexical, syntactic, semantics and constructions.

In the context of Indian languages, sarcasm detection is less explored due to the unavailability of benchmark resources for training and testing. Desai *et al.* [16] proposed a Support Vector Machine (SVM) based sarcasm detector for Hindi sentences. They used Hindi tweets as the dataset for training and testing using SVM classifier. In the absence of annotated datasets for training and testing, they converted English tweets into Hindi. Therefore, they focused on a similar set of features like emoticons and punctuation marks for sarcasm detection in English text. These methods when applied directly, are not suitable for the Hindi sarcastic tweets. Similarly, a context-based approach for sarcasm detection in Hindi tweets is proposed [19]. They exploit online Hindi news as a context to determine the sarcasm in tweets. They experimented using a small set of manually collected and annotated Hindi tweets.

III. Preliminaries

This section describes the predefined tools used in this article namely, Hindi POS tagging and Hindi SentiWordNet. The details are explained below:

A. POS Tagging

To identify the POS tag information in Hindi sentences, we have developed a Hidden Markov Model (HMM) based POS tagger [20]. It uses Indian Language (IL) standard tagset which consists of 24 tags [21]. The POS tagger is available on URL:http//www.taghindi.herokuapp.com. For example, POS tagging information of Hindi sentence क्या आपका नाम राम है? is क्या - WQ | आपका - PRP | नाम - NN | राम - NNP | है - VAUX | ? - SYM |.

B. Hindi SentiWordNet

To identify the sentiment of a Hindi tweet, we use a predefined list of Hindi SentiWordNet with polarity value. The SentiWordNet contains a total of 3014 annotated Hindi positive and Hindi negative words with possible synonyms. The SentiWordNet has five fields namely, POS tag, Synset ID (Hindi WN), Positive score, Negative score, and Related terms separated by comma. The list of Hindi SentiWordNet is available on URL:https://github.com/smadha/SarcasmDetector/tree/master/Hindi%20SentiWordNet.

IV. Proposed Scheme

This section describes the framework for sarcasm detection in Hindi tweets using the news as context as shown in Fig. 2. It starts with news collection from different news sources on Twitter followed by keyword extraction from each collected news to fetch the possible tweets related to the particular news. Next, every news from news corpus is fed to the Sarcasm Detection Engine (SDE) together with possible tweets from

[1] A natural Hindi tweet is a tweet that is available on Twitter in natural Hindi language unlike translated from English to Hindi.

[2] A timestamp is a news releasing date and time.

tweet corpus related to every news to detect the actual polarity of the tweets for corresponding news. In this article, the news is used as the context of the related tweets. If any tweet contradicts its context, then the tweet is classified as sarcastic.

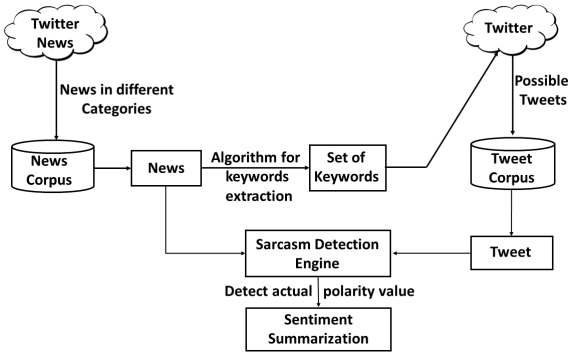

Fig. 2. System model for capturing and analysing sarcastic sentiment in Hindi tweets using Hindi news as context.

A. News and Tweet Collection

After browsing several Hindi news sources on Twitter, we have collected a total of around 2500 Hindi news manually from five different news sources as mentioned in Table I. This news is collected in five different categories as shown in Table I. The collected news consists of one liner news with a timestamp on very recent trending topics such as #EVMs, #EVMtampering, #Bahubali2, #khelratna, #BiharCM, #deputycmbihar *etc.* We intentionally omitted the news related to murder, rape, bomb blast, *etc.* We believe that sarcastic tweets will not be floated on grave topics. A sample Hindi news on EVMs is given in Fig. 3. In the preprocessing, we eliminated the redundant news. After preprocessing, the news corpus consists of a total of 1000 authenticated distinct news. The datasets of Hindi news is released on URL: https://github.com/sbharti1984/Hindi-News.

TABLE I
TWITTER NEWS SOURCE: HINDI NEWS AND NEWSPAPERS

Twitter News Sources	News Categories
आज तक @aajtak	Movies
ABP News हिंदी @ABPNewsHindi	Sports
BBC Hindi @BBCHindi	Celebrities
Dainik jagran @JagranNews	Entertainment
ZEE News Hindi @ZeeNewsHindi	Politics

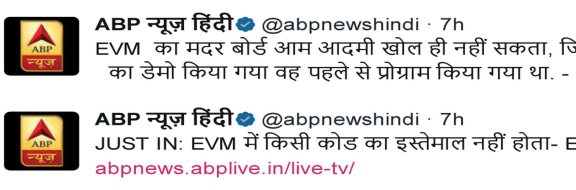

Fig. 3. A sample Hindi news on EVM.

To extract the important keywords of all the collected and processed news in news corpus, Algorithm 1 is used. Algorithm 1 takes authenticated news corpus as an input to extract an important set of keywords for every news in the news corpus to get the possible tweets based on the set of keywords. For every news, Algorithm finds POS tag information and check the presence of proper noun, verb and noun. If these tags are present, then the corresponding keywords are appended to the set of keywords. Here, our main focus is to extract the subject, object and verb of the news as important keywords. We assume that the tags, proper noun (NNP), verb (V), and noun (NN) will work as subject, verb, and object, respectively for the given news sentence. Thus, we extract the keywords of NNP, V, and NN tags.

Algorithm 1: *Important_Keywords_Extraction* (IKE)

Data: $dataset$:= Corpus of authenticated news ($\mathbb{C}2$)
Result: $classification$:= $\langle Set\ of\ Keywords, Tag \rangle$ for every news
Notation: NNP:Proper Noun, V: Verb, NN: Noun, NS: News Sentence, $\mathbb{C}$: Corpus, T: Tag, K: Keyword, NTS: News-wise Tagged Set, NKS: News-wise Set of Keywords, LoK: List of Keywords.
Initialization : $NKS = \{\ \phi\ \}$, $LoK = \{\ \phi\ \}$
while (NS *in* $\mathbb{C}2$) **do**
  NTS = find_POS_tag (NS)
  while (T *in* NTS) **do**
    if ($T == (NNP|V|NN)$) **then**
      $K \leftarrow$ Keyword[T]
    end
    $\langle NKS \rangle \leftarrow NKS \cup K$
  end
  $LoK \leftarrow LoK \cup \langle NKS \rangle$
end

Using the extracted keywords and similar trending topics, approximately 4000 Hindi tweets were collected from Twitter manually in the same timestamp. A sample of tweets on #EVMs is shown in Figure 4. The datasets of Hindi tweets are released on URL: https://github.com/sbharti1984/Hindi-Tweets.

Fig. 4. A sample Hindi tweets related to EVM.

B. Tweet Annotation

The collected Hindi tweets were distributed for annotation among three professionals in the Hindi language who are teachers and practitioners. They annotated these 4000 tweets manually to identify the tweet as sarcastic or not with the context of related news. After collecting the annotation results from all the individuals, we calculate the Inter-annotator agreement (IAA). To compute the IAA, Fleiss' Kappa coefficient [22] is used as it is more suitable when the number of

annotators is more than two. Fleiss' Kappa coefficient can be calculated using Equation 1. In this experiment, the attained IAA value is 0.89.

$$k = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \quad (1)$$

where,
$\bar{P} - \bar{P}_e$: gives the degree of agreement actually achieved above chance,
$1 - \bar{P}_e$: gives the degree of agreement that is attainable above chance.

The annotation result of 4000 tweets is used as ground truth to check the performance of the proposed system and error analysis. Among the 4000 tweets, 3000 tweets were used as observation set and remaining 1000 tweets were used as testing set. After verifying the annotation results of all the three annotators, the result is shown in Table II.

TABLE II
ANNOTATION RESULT OF 3000 HINDI TWEETS AS SARCASTIC OR NOT.

	#Tweets	sarcastic	non-sarcastic	ambiguous
Observation Set	3000	1036	1670	294
Testing Set	1000	370	573	57

After annotation, we observed that most of the sarcastic tweets in the observation set are contradicting with the context of related news. Therefore, we proposed a pattern-based sarcasm detection system using pattern "sarcasm as a contrast between a tweet and the context of its related news".

C. Sarcasm Detection Engine

This section describes the process of sarcasm detection in Hindi tweets as shown in Fig. 5. It takes a tweet along with related news as input and finds the POS tag information of both news and tweet. Then, sentiment identification algorithm is applied on tagged tweet file, and context identification algorithm is applied on the tweet as well as news tagged file to identify the sentiment and context of tweets and news respectively.

1) Sentiment Identification Algorithm: This section identifies the sentiment of a tweet. The sentiment value can be either positive, negative or neutral. The procedure for identifying sentiment value is given in Algorithm 2.

Algorithm 2 takes a corpus of Hindi tweets ($\mathbb{C}1$) and Hindi SentiWordNet($HSWN$) as input and produce the sentiment value of given tweet as negative, positive or neutral. For every tweet, it finds POS tag information and extracts the keyword value of tags noun, verb, adjective, and adverb. If these keywords are present in the list of Hindi SentiWordNet, then compute the negative and positive polarity value as SentiWordNet provides positive and negative polarity values of each word. Based on the maximum likelihood value of polarity, tweet classifies as negative or positive.

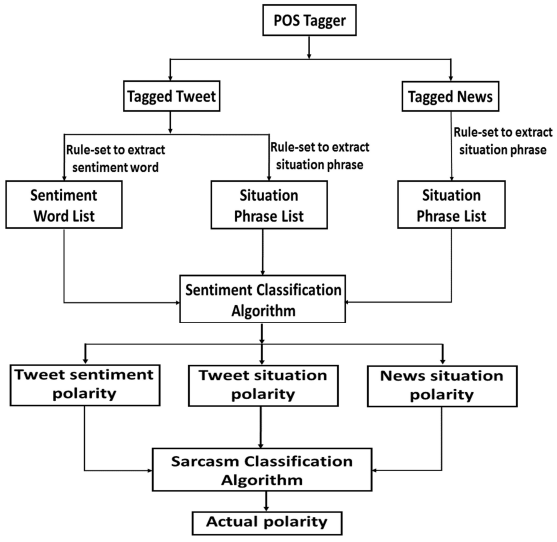

Fig. 5. Procedure for sarcasm detection in Hindi tweets.

Algorithm 2: Sentiment_Identification_Algorithm (SIA)

Data: $dataset$:= Corpus of Hindi Tweets ($\mathbb{C}1$), Hindi SentiWordNet ($HSWN$)
Result: $classification$:= Positive, Negative, Neutral
Notation: ADJ: Adjective, V: Verb, ADV: Adverb, NN: Noun, HT: Hindi Tweet, $\mathbb{C}$: Corpus, SW: Sentiment Word, K: Keyword, TTS: Tweet-wise Tagged Set, TP_s : Total Positive Score, TN_s : Total Negative Score.
Initialization : TP_s = { ϕ }, TN_s = { ϕ }
while (HT *in* $\mathbb{C}1$) **do**
  TTS = find_POS_tag (T)
  while (tag *in* TTS) **do**
    if ($tag == (ADJ|V|ADV|NN)$) **then**
      $SW \leftarrow$ Keyword[tag]
      if ($SW present\ in\ HSWN$) **then**
        P_s = Find_Positive_Polarity_Score (SW)
        N_s = Find_Negative_Polarity_Score (SW)
      end
      else
        Tweet is classified as neutral.
      end
    end
    else
      Tweet is classified as neutral.
    end
    $TP_s \leftarrow TP_s \cup P_s$
    $TN_s \leftarrow TN_s \cup N_s$
  end
  if ($TP_s > TN_s$) **then**
    Tweet is classified as positive.
  end
  else
    Tweet is classified as negative.
  end
end

2) Context Identification in Tweet and News: This section identifies the polarity of context for a tweet and the related news. The polarity value can be either positive or negative. The procedure for identifying context polarity is given in Algorithm 3.

Algorithm 3 takes Tweet Corpus ($\mathbb{C}1$), News Corpus ($\mathbb{C}2$) and Hindi SentiWordNet ($HSWN$) as input and produce the context polarity of tweet and news using HSWN as output. For every tweet and news, it finds POS tag information and extracts context phrase for both tweet and news using rule set

Algorithm 3: $Context\_Identification\_Algorithm\ (CIA)$

Input: Tweet Corpus ($\mathbb{C}1$), News Corpus ($\mathbb{C}2$), Hindi SentiWordNet ($HSWN$)
Output: Positive, Negative
Notation: ADJ: Adjective, V: Verb, ADV: Adverb, NN: Noun, T: Tweet, N: News, NTF: News tagged file, TTF: Tweet tagged file, NCP: News-wise context phrase, TCP: Tweet-wise context phrase, $BGTF$: Bigram tag file, $TGTF$: Trigram tag file
Initialization : $NCP = \{\phi\}$
while (T *in* $\mathbb{C}1$) **do**
  $TTF = Find\_POS\_Tag\ (T)$
  $BGTF_t = Find\_Bigram\_Tag\ (TTF)$
  $TGTF_t = Find\_Trigram\_Tag\ (TTF)$
  if $(BGTF_t == (ADV + NN)|(ADJ + V)|(NN + ADJ)|(ADJ + NN)|(NN + ADV)|(ADV + V))$ **then**
    $TCP \leftarrow TCP \cup Phrase[BGTF_t]$
  end
  else if $(TGTF_t == (ADV + ADJ + NN)|(V + ADJ + NN)|(V + ADV + ADJ)|(V + NN + NN)|(NN + V + NN)|(NN + NN + V))$ **then**
    $TCP \leftarrow TCP \cup Phrase[TGTF_t]$
  end
  while $(phrase\ in\ TCP)$ **do**
    while $(word\ in\ phrase)$ **do**
      if $(word\ present\ in\ HSWN)$ **then**
        P_s = Find_Positive_Polarity_Score $(word)$
        N_s = Find_Negative_Polarity_Score $(word)$
      end
    end
  end
  if $(TP_s > TN_s)$ **then**
    Tweet context is positive.
  end
  else
    Tweet context is negative.
  end
end
while (N *in* $\mathbb{C}2$) **do**
  $NTF = Find\_POS\_Tag\ (N)$
  $BGTF_n = Find\_Bigram\_Tag\ (NTF)$
  $TGTF_n = Find\_Trigram\_Tag\ (NTF)$
  if $(BGTF_n == (ADV + NN)|(ADJ + V)|(NN + ADJ)|(ADJ + NN)|(NN + ADV)|(ADV + V))$ **then**
    $NCP \leftarrow TCP \cup Phrase[BGTF_n]$
  end
  else if $(TGTF_n == (ADV + ADJ + NN)|(V + ADJ + NN)|(V + ADV + ADJ)|(V + NN + NN)|(NN + V + NN)|(NN + NN + V))$ **then**
    $NCP \leftarrow TCP \cup Phrase[TGTF_n]$
  end
  while $(phrase\ in\ NCP)$ **do**
    while $(word\ in\ phrase)$ **do**
      if $(word\ present\ in\ HSWN)$ **then**
        P_s = Find_Positive_Polarity_Score $(word)$
        N_s = Find_Negative_Polarity_Score $(word)$
      end
    end
  end
  if $(TP_s > TN_s)$ **then**
    News context is positive.
  end
  else
    News context is negative.
  end
end

given in Algorithm 3. Further, it finds the polarity value of each phrase using HSWN to identify the context polarity of the tweet and related news.

D. Sarcasm Detection Algorithm

In this article, the proposed approach for sarcasm detection in Hindi tweets is based on the pattern "sarcasm as a contradiction between a tweet and the context of its related news". The procedure for verifying the contradiction between a tweet and its related news context is given in Algorithm 4.

Algorithm 4: $Tweet\_Contradict\_Sentiment\_and\_Context$

Data: $dataset$:= Corpus of tweets ($\mathbb{C}1$), News corpus ($\mathbb{C}2$)
Result: $classification$:= sarcastic or non-sarcastic
Notation: TS: Tweet sentiment, T: Tweet, N: News, PTC: Tweet context polarity, PNC: News context polarity.
while (T *in* $\mathbb{C}1$) **do**
  TS = find_sentiment (T)
  PTC = find_context_polarity (T)
end
while (N *in* $\mathbb{C}2$) **do**
  NTC = find_context_polarity (T)
end
if $(PTC \neq NTC)$ **then**
  Tweets is classified as sarcastic.
end
else if $(PTC == NTC)\&\&(TS \neq PTC)$ **then**
  Tweets is classified as sarcastic.
end
else
  The tweet is non-sarcastic.
end

Algorithm 4 says, if the context polarity of a tweet is contradicted with context polarity of the related news then the tweet is classified as sarcastic. If the context polarity of a tweet and related news is same, but the sentiment of the tweet is contradicted then the tweet is classified as sarcastic. Otherwise, the tweet is not sarcastic.

V. Experimental Results

This section describes the experimental results of the proposed approach to identify sarcasm in Hindi tweets. There are four statistical parameters considered namely, $accuracy$, $precision$, $recall$ and $F1\text{-}measure$ to evaluate the performance of proposed approach. The $accuracy$ determines the performance of a system. A formula to calculate $accuracy$ is given in Equation 2.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_n + F_p} \tag{2}$$

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

$Precision$ and $recall$ are the other parameters which determine the performance of a system. Here, $precision$ determines how much relevant information system identified. Whereas, $recall$ determines how much identified information is relevant. The formula to ascertain $precision$ and $recall$ appeared in equations 3 and 4.

$$Precision = \frac{T_p}{T_p + F_p} \tag{3}$$

$$Recall = \frac{T_p}{T_p + F_n} \tag{4}$$

The value of precision and recall may vary application to application. For example, an application attains high precision but low recall. Similarly, another application attains low precision but high recall. To deal with this situation, one can rely on another statistical parameter *i.e.* $F1\text{-}measure$. It is a harmonic mean of $precision$ and $recall$. To obtain the $F1\text{-}measure$, a formula is given in Equation 5 .

$$F1\text{-}measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5}$$

A set of 1000 random Hindi tweets is used as a testing set to experiment. After the experiment, a confusion matrix of 1000 Hindi tweets is given in Table III for error analysis. To identify sarcasm in tweets, the proposed system identified 348 tweets correctly out of 370 sarcastic tweets. The ground truth of sarcastic and non-sarcastic tweets are given in Table II.

TABLE III
CONFUSION MATRIX FOR SARCASM DETECTION IN HINDI TWEETS.

Proposed approach	No. of tweets	T_p	T_n	F_p	F_n
Identifying sarcasm	1000	348	522	62	68

Using the confusion matrix given in Table III, the values of $precision$, $recall$, $F1\text{-}measure$ and $accuracy$, attained by the proposed approach for identifying sarcasm in tweets are given in Table IV. Finally, we made a comparison of the proposed approach with state-of-the-art approaches and is shown in Table V.

TABLE IV
ACCURACY, PRECISION, RECALL AND F1-MEASURE

Proposed approach	$Precision$	$Recall$	$F1\text{-}score$	$Accuracy$
Identifying sarcasm	0.848	0.836	0.842	0.87

TABLE V
COMPARISON OF PROPOSED APPROACH WITH SOME OF THE STATE-OF-THE-ART TECHNIQUES FOR SARCASM DETECTION IN HINDI TWEETS.

Study	$Precision$	$Recall$	$F1\text{-}score$	$Accuracy$
Desai *et al.*	0.732	0.674	0.705	0.714
Bharti *et al.*	0.736	0.717	0.726	0.794
Proposed Approach	**0.848**	**0.836**	**0.842**	**0.87**

VI. CONCLUSION

In the absence of sufficient dataset for training and testing, detection of sarcastic sentiment is a challenging task in Hindi. This article proposed a context-based pattern for sarcasm detection in Hindi tweets. Online News from Twitter news sources is utilized as the context of the tweets in the same timestamp. The proposed approach attains an accuracy of 87%. The proposed approach outperforms the state-of-the-arts techniques for sarcasm detection in Hindi tweets.

REFERENCES

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

[2] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of conference on Empirical methods in natural language processing*, vol. 10, no. 1. ACL, 2002, pp. 79–86.

[3] C. Liebrecht, F. Kunneman, and A. van den Bosch, "The perfect solution for detecting sarcasm in tweets# not," in *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. New Brunswick, NJ: ACL, 2013, pp. 29–37.

[4] R. González-Ibánez, S. Muresan, and N. Wacholder, "Identifying sarcasm in twitter: a closer look," in *Proceedings of the 49th Annual Meeting on Human Language Technologies*. ACL, 2011, pp. 581–586.

[5] F. Barbieri, H. Saggion, and F. Ronzano, "Modelling sarcasm in twitter a novel approach," *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 50–58, 2014.

[6] 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 *Proceedings of the conference on empirical methods in natural language processing*, 2013, pp. 704–714.

[7] S. K. Bharti, K. S. Babu, and S. K. Jena, "Parsing-based sarcasm sentiment recognition in twitter data," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. ACM, 2015, pp. 1373–1380.

[8] 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*. ACL, 2010, pp. 107–116.

[9] F. Kunneman, C. Liebrecht, M. van Mulken, and A. van den Bosch, "Signaling sarcasm: From hyperbole to hashtag," *Information Processing & Management*, vol. 51, no. 4, pp. 500–509, 2015.

[10] 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*. ACM, 2015, pp. 97–106.

[11] S. Bharti, B. Vachha, R. Pradhan, K. Babu, and S. Jena, "Sarcastic sentiment detection in tweets streamed in real time: a big data approach," *Digital Communications and Networks*, vol. 2, no. 3, pp. 108–121, 2016.

[12] S. K. Bharti, R. Pradhan, K. S. Babu, and S. K. Jena, "Sarcastic sentiment detection based on types of sarcasm occurring in twitter data," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 13, no. 4, pp. 89–108, 2017.

[13] M. Parkvall, "Världens 100 största språk 2007," *The World's*, vol. 100, 2007.

[14] Language and Culture, "Top 30 languages by number of native speakers," 2015. [Online]. Available: http://www.vistawide.com/languages/top-30-llanguages.htm/

[15] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proceedings of Language Resources and Evaluation Conference*, 2006, pp. 417–422.

[16] N. Desai and A. D. Dave, "Sarcasm detection in hindi sentences using support vector machine," *International Journal*, vol. 4, no. 7, pp. 8–15, 2016.

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

[18] 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*, 2014, pp. 459–471.

[19] S. Bharti, S. Korra, and S. Jena, "Harnessing online news for sarcasm detection in hindi tweets." *Proceeding of PReMI 2017 to be published by LNCS*, 2017.

[20] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, "A practical part-of-speech tagger," in *Proceedings of the third conference on Applied natural language processing*. Association for Computational Linguistics, 1992, pp. 133–140.

[21] A. Bharati, R. Sangal, D. M. Sharma, and L. Bai, "Anncorra: Annotating corpora guidelines for pos and chunk annotation for indian languages," *LTRC-TR31*, 2006.

[22] J. L. Fleiss, J. Cohen, and B. Everitt, "Large sample standard errors of kappa and weighted kappa." *Psychological Bulletin*, vol. 72, no. 5, p. 323, 1969.