Sarcasm Detection in Hindi Tweets Streamed in Real Time: A Study in the Social Media Paradigm

Santosh Kumar Bharti, National Institute of Technology Rourkela, India
Rahul Raman, National Institute of Technology Rourkela, India
Korra Sathya Babu, National Institute of Technology Rourkela, India
Sanjay Kumar Jena, National Institute of Technology Rourkela, India

Sentiment analysis is the way of finding ones' opinion towards any specific target. Sarcasm is the special types of sentiment which infer the opposite meaning of what people are trying to 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 is becoming the most challenging task. Researchers are unable to touch expected accuracy level for sentiment analysis. Therefore, an automated sarcasm detector is required for textual data. In recent times, many researchers have worked on this area and proposed many sarcasm detection techniques. These techniques are designed to detect sarcasm on the data scripted in English since it is the most popular language in social networking groups. Moreover, due to its popularity, plenty of resources are available in English. However, parallel research for sarcasm detection on different Asian languages like Hindi, Telugu, Tamil, Urdu, and Bengali are not available. One of the reasons for the less exploration of these languages for sentiment analysis is the lack of available databases even though they are popular in a large networked society. Due to the lack of availability of annotated corpora and its complex morphology, identification of sentiments (especially sarcastic sentiment) in these languages becomes challenging task. In this paper, we have proposed a framework for creating the list of positive and negative Hindi words or phrases for sentiment analysis in Hindi tweets. Further, it can also be used to detect sarcastic Hindi tweets. The backbone of the whole framework is online news which is considered as the context of the tweets.

CCS Concepts: •**Information systems** → **Information integration; Data analytics; Online analytical processing;**

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: e-Newspapers, Online News, Sarcastic, Sentiment, Social Media, Tweets

ACM Reference Format:
Santosh Kumar Bharti, Rahul Raman, Korra Sathya Babu, and Sanjay Kumar Jena. Sarcastic Sentiment Detection in Hindi Tweets: A Study in the Social Media Paradigm. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* V, N, Article A (January YYYY), 13 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Online companion has gained tremendous momentum in recent times for business, politics, entertainment, etc. Social media such as Twitter, Facebook, WhatsApp, etc. is the popular medium for online companion and attaining the response from worldwide. These responses include ones sentiment or opinion towards any specific target such as individuals, events, topics, products, organizations, services, etc. The sentiment is nothing but an opinion of any individual towards a specific target. It may be either positive or negative. Manual extraction of the sentiment from social media is a tedious job for individuals or organizations. There is a need for an automated system which will be capable of providing sentiment value from social media in real-time.

Sentiment analysis is a Natural Language Processing (NLP) task that deals with finding the orientation of an opinion in a piece of text about a topic [Pang et al. 2002]. The major challenges involve

Author's addresses: Santosh Kumar Bharti, Rahul Raman, Korra Sathya Babu, and Sanjay Kumar Jena, Computer Science Department, National Institute of Technology Rourkela-769008, Odisha, India.

DOI: http://dx.doi.org/10.1145/0000000.0000000

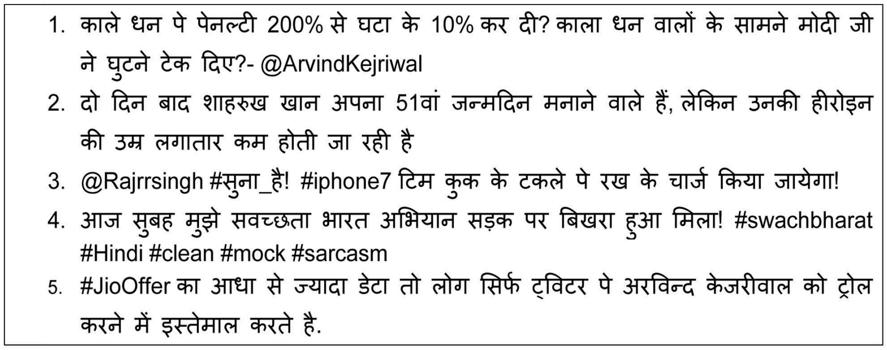
for sentiment analysis is the presence of sarcasm in the dataset. Due to the sarcastic text, most of the existing systems for sentiment analysis fails in detecting the actual sentiment value. Sarcasm is a special kind of sentiment that usually flip the orientation of the opinion in a given piece of text. According to Macmillan English Dictionary, sarcasm is "the activity of saying or writing the opposite of what you mean, or of speaking in a way intended to make someone else feel stupid or show them that you are angry". 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 challenging. Sometimes, human beings feel difficulty to understand sarcasm in text. To get a better accuracy in sentiment analysis, one needs to consider sarcasm as well.

Sarcasm detection is another NLP task that deals with finding the actual orientation of the opinion such as actual positive, actual negative or sarcastic. The sarcastic sentence usually looks positive, but overall meaning becomes negative due to the presence of sarcasm. Therefore, an automated system is required for sentiment analysis which is capable of identifying the actual sentiment or sarcastic sentiment. In recent past, many researchers have focused on sarcasm detection and proposed automatic sarcasm detector in text [Liebrecht et al. 2013; González-Ibánez et al. 2011; Barbieri et al. 2014; Riloff et al. 2013; Bharti et al. 2015; Davidov et al. 2010; Kunneman et al. 2015; Rajadesingan et al. 2015; Bharti et al. 2016]. These systems are developed for sarcasm detection in text scripted in English. English is the most popular language across worldwide and in this domain several resources are freely available for research. Therefore, majority of researchers have preferred English datasets 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 [Parkvall 2007]. It has 490 million speakers across the world and majority of them are from India [Language and Culture 2005]. It is widely used for speaking in countries like India, Mauritius, Fiji, Suriname, Guyana, Trinidad & Tobago and Nepal [Esuli and Sebastiani 2006]. 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 [Dalal et al. 2006; Awasthi et al. 2006; Singh et al. 2006; Mishra and Mishra 2011; Mall and Jaiswal 2011; Garg et al. 2012; Joshi et al. 2013; Narayan et al. 2014], sentiment analysis [Prasad et al. 2015; Sharma et al. 2015; Mittal et al. 2013; Mulatkar 2014; Joshi et al. 2010] had been already developed.

Mining sarcastic sentiment in Hindi text comes with their share of issues and challenges. Hindi is morphologically rich and is a free order language as compared to English, which adds complexity while handling the user-generated content. The scarcity of resources for the Hindi language brings challenges ranging from collection to generation of datasets. So far, very few research had been reported in low resourced languages [Lunando and Purwarianti 2013; Liu et al. 2014; Desai and Dave 2016]. In the Hindi language, the only reported work is [Desai and Dave 2016] to the best of our knowledge. They used Hindi tweets for analysis which were translated from English tweets. They haven‘t used natural Hindi sentences or tweets for analysis.

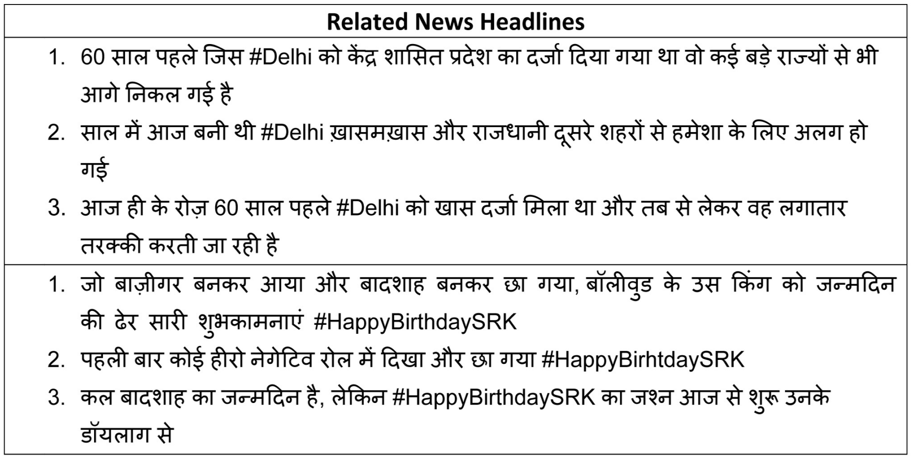
In this paper, we proposed a novel approach to detect sarcasm in Hindi tweets that are taken from Twitter as shown in Fig. 1. These tweets are natural Hindi sentences and to identify sarcasm from it, we have used news data as the context to classify a tweet as sarcastic or not. In this approach, we collected one liner related news as shown in Fig. 2 from various Hindi e-Newspapers and several online news sites in different categories as shown in Table I. Next, we compared all the related news to extract all the meaningful keywords from the related news (redundant keywords are eliminated). The extracted keywords are used to summarized several related news into one authenticated news. Further, we used same extracted keywords to collect tweets manually from Twitter.



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.

For identification, we map each tweet to one of the news in summarized news corpus and compare the sentiment value of both summarized news and tweet. If sentiment value of the tweet contradicts with corresponding news sentiment, the tweet is classified as sarcastic; otherwise, it is not sarcastic.



Related News Headlines

1. 60 साल पहले जिस #Delhi को केंद्र शासित प्रदेश का दर्जा दिया गया था वो कई बड़े राज्यों से भी आगे निकल गई है
2. साल में आज बनी थी #Delhi ख़ासमख़ास और राजधानी दूसरे शहरों से हमेशा के लिए अलग हो गई
3. आज ही के रोज़ 60 साल पहले #Delhi को खास दर्जा मिला था और तब से लेकर वह लगातार तरक्की करती जा रही है

1. जो बाज़ीगर बनकर आया और बादशाह बनकर छा गया, बॉलीवुड के उस किंग को जन्मदिन की ढेर सारी शुभकामनाएं #HappyBirthdaySRK
2. पहली बार कोई हीरो नेगेटिव रोल में दिखा और छा गया #HappyBirhtdaySRK
3. कल बादशाह का जन्मदिन है, लेकिन #HappyBirthdaySRK का जश्न आज से शुरू उनके डॉयलाग से

Fig. 2: A sample Hindi related news headlines.

Rest of the paper is organized as follows: Section 2 describes related work. The proposed scheme is discussed in Section 3. Analysis of the results are given in Section 4 and conclusion of the article is drawn in Section 5.

2. RELATED WORK

In recent times, several authors have already explored and proposed their sarcasm detector for text scripted in the English language [Liebrecht et al. 2013; González-Ibánez et al. 2011; Barbieri et al. 2014; Riloff et al. 2013; Bharti et al. 2015; Davidov et al. 2010; Kunneman et al. 2015; Rajadesingan et al. 2015; Bharti et al. 2016]. This paper explored the previous concept of identifying sarcasm in low resource languages such as Hindi, Tamil, Telugu, Urdu, Arabic, Indonesian, Mandarin, etc., [Lunando and Purwarianti 2013; Liu et al. 2014; Desai and Dave 2016] where availability of datasets are very rare.

Table I: News Source: Hindi e-Newspapers, Hindi TV News sites

News Sources	**News Categories**
Dainik Bhaskar	Sports
AajTak News	Movies
ABP News	Business
Wah Cricket	Politics
BBC Hindi	Celebrities
Hindustan	National
NavBharat Times	Literature
Dainik Jagran	Technology
ZEE News Hindi	Environment
Amar Ujala	
Hindustan	
Daily Hindi News	

A sarcasm detector was proposed [Lunando and Purwarianti 2013] 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 the high tendency to be sarcastic. Similarly, a multi-strategy ensemble classification algorithm was proposed [Liu et al. 2014] for Chinese social media to identify sarcasm in Chinese tweets using a comprehensive sarcasm feature set including lexical, syntactic, semantics and constructions.

In Hindi, several authors have proposed automatic sentiment analyzers [Prasad et al. 2015; Sharma et al. 2015; Mittal et al. 2013; Mulatkar 2014; Joshi et al. 2010] to identify the sentiment in Hindi texts, tweets and reviews. For sarcasm detection in Hindi, only reported work is [Desai and Dave 2016]. They have used Hindi tweets as the dataset for training and testing using support vector machine classifier. They have focused on features like emoticons and punctuation marks to identify sarcastic tweets. Similarly, an insult detection in Hindi text was proposed [Dalal et al. 2014] using logistic regression and support vector machine classifiers. They have used n-grams, negation and second person as features to build the feature vector to train these classifiers.

3. PROPOSED SCHEME

This section describes the framework for sarcasm detection in Hindi tweets using news as context as shown in Fig. 3. It starts with tweets collection followed by keyword extraction from collected tweets and news collection. Then using extracted keywords from every tweet, selection of appropriate news for every tweet. Next, it fed both tweet and its corresponding news to Sarcasm Detection Engine (SDE) to detect actual polarity of given tweet.

3.1. Tweets Collection

In this step, all the meaningful keywords are extracted from every news headlines in the news corpus. Next, the extracted keywords are used to summarize all the related news from different news sources to one authenticated news. For keywords extraction and news summarization, we followed [Thomas et al. 2016] procedure for automatic keyword extraction and summarization. The purpose of summarization is to eliminate the redundancy in the related news and make a corpus of single authenticated news of several related news. Therefore, after summarization, news corpus contains 500 authenticated news. We used this summarized authenticated news as the base information for sarcasm detection in Hindi tweets.

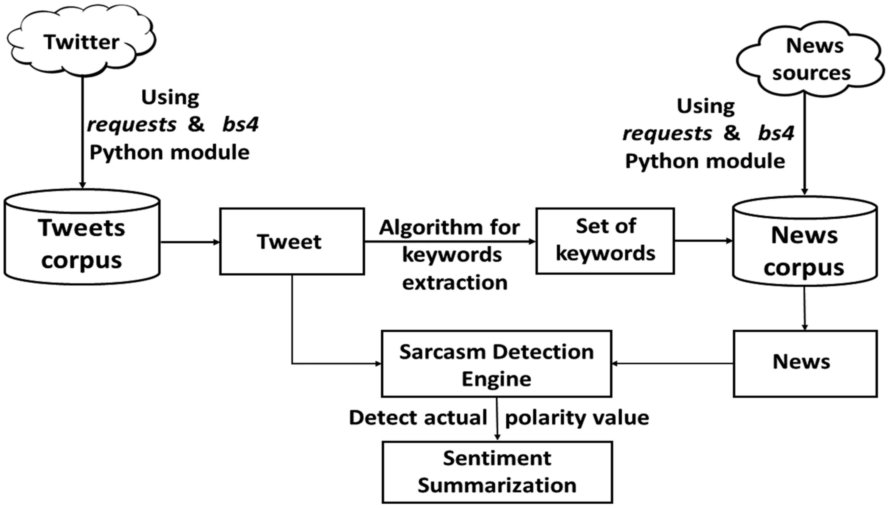

Twitter
Using *requests* & *bs4* Python module
News sources
Using *requests* & *bs4* Python module
Tweets corpus
Tweet
Algorithm for keywords extraction
Set of keywords
News corpus
Sarcasm Detection Engine
News
Detect actual polarity value
Sentiment Summarization

Fig. 3: System model for capturing and analysing sarcasm sentiment in Hindi tweets using Hindi News as contaxt.

3.2. Keyword Extraction Algorithm

The meaningful keywords extracted from news corpus are used to collect tweets from Twitter. In this work, we have collected around 5000 Hindi tweets manually from Twitter. Sample Hindi sarcastic tweets are shown in Fig. 1.

ALGORITHM 1: Keywords_Extraction_Algorithm

```
Data: $dataset$ := Corpus of authenticated news ($\mathbb{C}$)
Result: $classification$:= $\langle Setofkeywords, Tag \rangle$ for every news
Notation: $ADJ$: Adjective, $V$: Verb, $ADV$: Adverb, $NN$: 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 $\mathbb{C}$ do
    $NTS$ = find_POS_tag ($NS$)
    while $T$ in $NTS$ do
        if ($T == (ADJ||V||ADV||NN)$) then
            $K \leftarrow$ Keyword[$T$]
        end
        $\langle NKS \rangle \leftarrow NKS \cup K$
    end
    $LoK \leftarrow LoK \cup \langle NKS \rangle$
end
```

3.3. News Collection

After browsing several e-News websites, we have collected a total of around 6000 one liner news in different categories such as sports, movies, business, politics, etc. from six different e-Newspapers and six different online news sources as mentioned in Fig. **??**. The news corpus contains almost 500 related news headlines from each e-Newspapers and online news sites between November 2016 to December 2016. Few sample related news headlines are given in Fig. 2.

3.4. Sarcasm Detection Engine

This section describes the proposed approach for detecting sarcasm in tweets. It starts with annotation of sentiment value in summarized news corpus followed by identifying sentiment value in Hindi tweets based on corresponding summarized news. Finally, it detects sarcastic tweets based on the comparison of sentiments of both news and tweets.

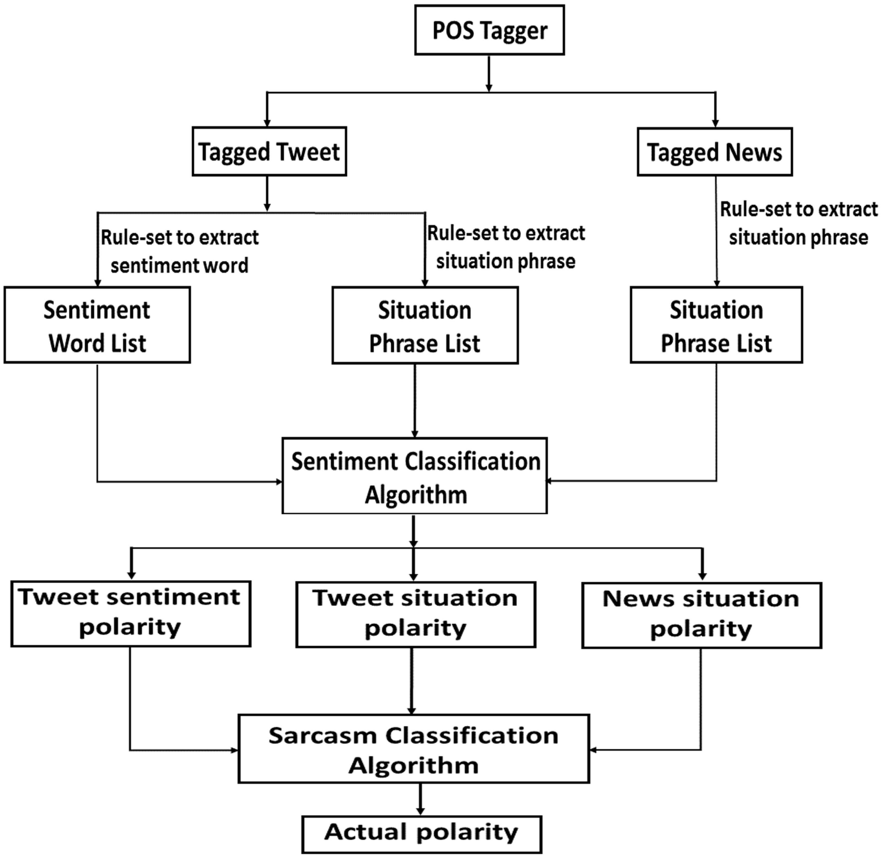


Fig. 4: Procedure of Data Collection.

3.4.1. POS Tagging. list[1]

3.4.2. Sentiment Word and Situation Phrase Generation.

3.4.3. Sentiment Classification Algorithm. For sentiment annotation, we have taken help from four annotators namely, Sonali Agarwal, Ashmita Poddar, Arvind Yadav, and Manoj Kumar Gupta for providing sentiment annotation for summarized news corpus. They have annotated the sentiment value as positive, negative and neutral for the summarized news. A sample annotated summarized news are shown in Fig. **??**.

[1] https://www.ethnologue.com/statistics/size

ALGORITHM 2: $Extraction\_of\_Phrases\_for\_SPNet\_Generation(EPSG)$

```
Input: Corpus of Telugu Sentences ($\mathbb{C}$)
Output: List of bigram and trigram phrases
Notation: $ADJ$: Adjective,$V$: Verb, $ADV$: Adverb, $NN$: Noun, S: sentence, $\mathbb{C}$: corpus, $TF$: tagged file, $T$:
  tag, $SPS$: Sentence-wise set of phrase, $PF$: Phrase file
Initialization : $PF = \{\emptyset\}$
while $S$ in $\mathbb{C}$ do
    $TF = Find\_POS\_Tag\ (S)$
    while $T$ in $TF$ do
        if (Pattern of (ADV +NN) || (ADJ +V ) || (NN +ADJ) || (ADJ +NN) || (NN + ADV ) || (ADV + V ) ||
          (ADV + ADJ + NN) || (V + ADJ + NN) || (V + ADV + ADJ) || (V + NN + NN) || (NN + V + NN) ||
          (NN + NN + V) is Matched) then
            $\langle PatSet\rangle = Extract\_matched\_pattern\_phrase\ (TF)$
            $\langle SPS\rangle = Phrase\ [PatSet]$
        end
        else
            Input Sentence is Neutral
        end
    end
    $PF \leftarrow PF \cup \langle SPS\rangle$
end
```

ALGORITHM 3: Sentiment_Identification_Algorithm

```
Data: $dataset$ := Corpus of authenticated news ($\mathbb{C}$)
Result: $classification$:= $\langle Setofkeywords, Tag\rangle$ for every news
Notation: $ADJ$: Adjective, $V$: Verb, $ADV$: Adverb, $NN$: 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 $\mathbb{C}$ do
    $NTS$ = find_POS_tag ($NS$) while $T$ in $NTS$ do
        if ($T == (ADJ||V||ADV||NN)$) then
            $K \leftarrow$ Keyword[$T$]
        end
        $\langle NKS\rangle \leftarrow NKS \cup K$
    end
    $LoK \leftarrow LoK \cup \langle NKS\rangle$
end
```

3.5. Sarcasm Detection Algorithm

In Hindi tweets, the identifying sentiment is very tough as the predefined list of negative and positive words are unavailable. Therefore, this paper proposed a novel approach for creating the list of positive and negative Hindi words or phrase to identify sentiment in tweets based on manual sentiment annotated news as ground truth as shown in Fig. **??**.

To create the list of negative and positive words, one need to analyze the sentiment value of the annotated summarized news. If the sentiment value of the news is positive, then the list of meaningful keywords (in the same sequence) of corresponding news is appended to the list of positive words. Otherwise, append it to the list of negative words. Fig. **??** describes the step-wise procedure for identifying sentiment value in Hindi tweets. To identify the sentiment of the particular tweet, one need to map a particular tweet to its corresponding news in annotated summarized news corpus. Then, extract meaningful keywords from both tweet and the corresponding news. Next, we check the pattern and sequence of extracted keywords in both news and tweet and compare it. Due to the manual annotation, the sentiment value of news along the extracted keywords sequence is known.

ALGORITHM 4: $Sentiment\_Classification\_using\_SPNet$

```
Input: Testing set of Telugu sentences ($\mathtt{C}1$),$SPNet$
Output: $classification := positive, negativeorneutral$
Notation: $ADJ$: Adjective, $V$ : Verb, $ADV$ : Adverb, $NN$: Noun, $S$: sentence, $T$: Tag, $TF$: Tagged file,
  $PatSet$: Pattern set, $SPS$: Sentence-wise set of phrases, ($\mathtt{C}1$): Testing set, $UKPL$ : Unknown phrase list,
  $SC$: Sentiment score
Initialization : $UKPL$ = {∅}
while $S$ in ($\mathtt{C}1$) do
    $TF = Find\_POS\_Tag\ (S)$
    $countP = 0$
    while $T$ in $TF$ do
        if (Pattern of (ADV +NN) || (ADJ +V ) || (NN +ADJ) || (ADJ +NN) || (NN + ADV ) || (ADV + V ) ||
          (ADV + ADJ + NN) || (V + ADJ + NN) || (V + ADV + ADJ) || (V + NN + NN) || (NN + V + NN) ||
          (NN + NN + V) is Matched) then
            $\langle PatSet\rangle = Extract\_matched\_pattern\_phrase\ (TF)$
            $\langle SPS\rangle = Phrase\ [PatSet]$
        end
    end
    $N_c = 0$, $P_c = 0$, $Neu_c = 0$
    while (phrase in $SPS$) do
        check the presence of the phrase in SPNet.
        if (phrase present in positive list ) then
            $P_c \leftarrow P_c + 1$
        end
        else if (phrase present in negative list ) then
            $N_c \leftarrow N_c + 1$
        end
        else if (phrase present in neutral list ) then
            $Neu_c \leftarrow Neu_c + 1$
        end
    end
    if ( $CountP$ == $N_c$) then
        Given sentence is classified as negative.
    end
    else if ( $CountP$ == $P_c$ ) then
        Given sentence is classified as positive.
    end
    else if ( $CountP$ == $Neu_c$) then
        Given sentence is classified as neutral
    end
    else if (($P_c > 0$) && ($Neu_c > 0$) && ($N_c == 0$)) then
        Given sentence is classified as positive.
    end
    else if (($N_c > 0$) && ($Neu_c > 0$) && ($P_c == 0$)) then
        Given sentence is classified as negative.
    end
    else if (($P_c > 0$) && ($N_c > 0$) &&($Neu_c == 0$)) then
        $SC = Find\_Sentiment_score\ (SPS, PatSet)$
        if ($SC > 0$) then
            Given sentence is classified as positive
        end
        else if ($SC < 0$) then
            Given sentence is classified as negative
        end
        else
            Given sentence is classified as ambiguous.
        end
    end
    else
        $UKPL \leftarrow UKPL \cup SPS$
    end
end
```

```
ALGORITHM 5: $Sentiment\_Score\_Calculation$
Input: $SPS$, $PatSet$, $SPNet$
Output: $Sentiment\ score$
Notation: $S$: sentence, $SPS$: sentences-wise set of phrases, $SC$: Sentiment score, $ToP$: Tags of Phrase
Initialization : $Dict = f(ADV : 2); (ADJ : 3); (NN : 1); (V : 1)$, $W_p = \{0\}$ , $W_n = \{0\}$
while $(phrase\ in\ SPS)$ do
    $ToP = Extract\_Tagset\ (PatSet)$
    if phrase is present in positive list of either bigram or trigram then
        $W = argmax_{j\epsilon Dict}(ToP[j])$
        if $W > W_n$ then
            $W_n = W$
        end
    end
    else
        $W1 = argmax_{j\epsilon Dict}(ToP\,[j])$
        if $W1 > W_n$ then
            $W_n = W$
        end
    end
end
$SC = W_p - W_n$
```

```
ALGORITHM 6: Context_Identification_Algorithm
Data: $dataset$ := News sentence (NS)
Result: $classification$:= Context identification
Notation: $ADJ$: Adjective, $V$: Verb, $ADV$: Adverb, $NN$: Noun, $NS$: News Sentence, $T$: Tag, $PatSet$:
  Pattern set, $NTS$: News-wise Tagged Set, $NPS$: News-wise set of phrase, $LoK$: List of Keywords.
$NTS$ = find_POS_tag ($NS$)
while $T\ in\ NTS$ do
    if $(Pattern\ of(V+NN)||(ADJ+NN)||(ADJ+V)||(ADV+NN)||(ADV+V)||(ADV+ADJ+NN)||((V+ADJ+NN)||((V+ADV+ADJ))$ matched then
        $\langle PatSet\rangle = Extract\_matched\_pattern\_phrase\ (NTS)$
        $\langle NPS\rangle \leftarrow Phrase[PatSet]$
    end
end
```

Therefore, for comparison, if pattern and sequences of both extracted keywords are same then we predict the tweet sentiment value is same as the corresponding news. If the pattern does not match, then we use the method of [Joshi et al. 2010] for sentiment analysis and store the sentiment value along with corresponding tweets for further use.

3.6. Sarcasm Detection

(1) Contradiction between sentiment and context.
(2) Tweet contradict time dependent facts.

3.7. Contradiction between sentiment and context

3.8. Tweet contradict time dependent facts

Tweets contradicting with time-dependent facts (TCTDF) are based on temporal facts. In this approach, time-dependent facts (may change over certain time period) are used as a feature to identify sarcasm in tweets as shown in Algorithm 8. For an instance, ‘@MirzaSania becomes world number one. Great day for Indian tennis’ is a time-dependent fact sentence. May be after sometime, someone else will be number one tennis player. The newspaper headlines is used as a corpus for

ALGORITHM 7: Tweet_contradict_sentiment_and_context

```
Data: $dataset$ := Corpus of tweets ($\mathtt{C}$), News corpus ($\mathtt{C}1$)
Result: $classification$:= sarcastic or not sarcastic
Notation: $TS$: Tweet sentiment, $T$: Tweet, $NKS$: News-wise set of keywords, $NS$: News sentence, $\mathtt{C}$:
  Corpus, $NC$: News context.
while $T$ in $\mathtt{C}$ do
    $TS$ = find_sentiment ($T$)
    $\langle NKS \rangle$ = find_keywords ($T$)
    $NS$ = find_news_using_NKS ($\mathtt{C}1$)
    $NC$ = find_context ($NS$)
    if ($TS == NC$) then
        Given tweets is not sarcastic.
    end
    else
        Given tweets is sarcastic.
    end
end
```

ALGORITHM 8: Tweet_contradict_time_dependent_facts

```
Data: $dataset$ := Corpus of time-dependent facts.
Result: $classification$:= A $\langle Key, Value \rangle$ pair
Notation: $S$: Subject, $V$: Verb, $O$: Object, $T$: Tweets, $C$: Corpus, $PF$: Parse File, $TWP$: Tweet Wise Parse
  Phrase, $TDFF$: Time Dependent Fact File.
Initialization : $PF$ = { $\phi$ }, $TDFF$ = { $\phi$ }
while $T$ in $C$ do
    $p$ = find_parsing ($T$) $PF \leftarrow PF \cup p$
end
while $TWP$ in $PF$ do
    $S$ = find_subject ($TWP$)
    $V$ = find_verb ($TWP$)
    $O$ = find_object ($TWP$)
    Key $\leftarrow$ (S + V)
    Value $\leftarrow$ (O + TS)
    $UFF \leftarrow \langle Key, Value \rangle$
end
```

time-dependent facts. The Algorithm **??** use newspaper headlines as an input corpus and generates a list of $\langle key, value \rangle$ pair for every headlines in the corpus. To generate $\langle key, value \rangle$ pair, it finds triplet of (subject, verb, and object) values according to Rusu_Triplets [**?**] method for every sentence. Further, it combines subject and verb together as key and combine object and time-stamp as value. Time-stamp is the news headline date. The $\langle key, value \rangle$ pair for the sentence ‘Wow, Australia won the cricket world cup again in 2015’ is $\langle (Australia, won), (cricketworldcup, 2015) \rangle$.

3.9. Sarcasm Detection

In this paper, the proposed approach for sarcasm detection in Hindi tweets is based on the sentiment values of both the tweets and its corresponding news in annotated summarized news corpus as shown in Fig. **??**.

The approach of sarcasm detection is very simple. Here, if the sentiment values of both input tweet and its corresponding news are same, then the tweet is not sarcastic. Otherwise, the tweet is sarcastic. In this situation, the news acts as a context of the tweet when it is posted by some Twitter user. If the sentiment value of that tweet is contradicting the news fact, it means, it is written intentionally sarcastic.

In Algorithm 8, steps 1 to 4 find parsing phrases such as noun phrase (NP), verb phrase (VP), adverb phrase (ADVP), etc. and append it to phrase file (PF). Steps 6 to 8 extract triplets (subject, verb, object) from parse phrase. Steps 9 to 11 forms $\langle key, value \rangle$ pair using these triplets where "key" contains the subject (S) and verb (V) while "value" contains an object (O) and timestamp (TS). Hence, Algorithm 8 gives a learned list of time-dependent facts in the form of $\langle key, value \rangle$ pair.

Identifying sarcasm in tweets using time-dependent facts is similar to TCUF as shown in Algorithm 8. The only difference is in value of $\langle key, value \rangle$ pair. While matching $\langle key, value \rangle$ pair of testing tweet with $\langle key, value \rangle$ pair in file to identify sarcasm using TCTDF approach, one need to match object as well as time-stamp together as value. If both matches then, current testing tweet is not sarcastic else sarcastic.

4. RESULTS

This section describes the performance of the proposed approach to identify sarcasm in Hindi tweets. We started with the description of experimental setup followed by experimental results and its analysis.

4.1. Experimental Environment

The algorithms in proposed approach is deployed in a machine with configuration, Intel Xeon E5-2620 (6 core, v3 @ 2.4 GHz) processor with 4 GB RAM and minimum 20 GB free memory space. The operating system used as Ubuntu-14.04 $\times$ 64. Python 2.7. is installed for implementation. Python is an interpreted language which best suite for NLP task as plenty of the NLP packages are available freely such as NLTK, TextBlob, etc. It has a strong capability of handling string manipulation tasks.

4.2. Experimental Results

There are four statistical parameters that are considered namely, $accuracy$, $precision$, $recall$ and $f - measure$ to evaluate the performance of the purposed approach. The formula to ascertain $accuracy$, $precision$, $recall$ and $f - measure$ are given in equations 1, 2, 3, and 4 respectively.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + T_p + F_p} \tag{1}$$

$$Precision = \frac{T_p}{T_p + F_p} \tag{2}$$

$$Recall = \frac{T_p}{T_p + F_n} \tag{3}$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

where,
T_p = True positive, T_n = True negative, F_p = False positive, F_n = False negative
To test the performance of the proposed approach, we have experimented on 500 random tweets from collected Hindi tweets corpus. To annotate the sarcastic tweets, four annotators used to identify sarcastic tweets, and the results of annotators are used as ground truth for testing. A confusion matrix for predicting the sentiment value and identifying sarcasm in 500 tweets are given in Table II. To predict sentiment, if the pattern of extracted keywords of the summarized news doesn‘t match the pattern of extracted keywords of the tweet is treated as true negative (T_n). The proposed system doesn‘t predict the sentiment value for 119 tweets. The proposed approach predicted the sentiment

value correctly for 307 tweets and predicted incorrectly for 74 tweets. Similarly, for identifying sarcasm in tweets, proposed system identified 451 sarcastic tweets correctly and 49 tweets incorrectly out of 500 tweets as shown in Table II.

Table II: Confusion matrix.

Proposed approach	No. of tweets	T_p	T_n	F_p	F_n
Predicting sentiment	500	307	119	33	41
Identifying sarcasm	500	157	294	21	28

On the basis of the confusion matrix, the values of precision, recall, F-measure and accuracy attained by the proposed approach for predicting the sentiment value and identifying sarcasm in tweets are given in Table III.

Table III: Confusion matrix.

Proposed approach	$Precision$	$Recall$	$F - measure$	$Accuracy(\%)$
Predicting sentiment	0.902	0.882	0.891	85.2
Identifying sarcasm	0.882	0.848	0.865	90.2

5. CONCLUSION

In the absence of sufficient dataset for training and testing, detection of sarcastic sentiment is a challenging task in Hindi. This paper proposed a framework for creating the list of positive and negative Hindi words for identifying the sentiment value in Twitter data. Further, it deployed the sentiment value for sarcasm detection in tweets. The backbone of the whole framework is online news. We assumed the news are the context to identify sarcastic sentiment in tweets. The proposed approach attains very good accuracy level as there is not much-reported work available for sarcasm detection in Hindi.

ACKNOWLEDGMENTS

The authors would like to thank Sonali Agarwal, Ashmita Poddar, Arvind Yadav, and Manoj Kumar Gupta for providing sentiment annotation of our collected data set. All the annotators belong to the state of Bihar and Rajasthan, and their mother tong is Hindi.

REFERENCES

Pranjal Awasthi, Delip Rao, and Balaraman Ravindran. 2006. Part of speech tagging and chunking with hmm and crf. *Proceedings of NLP Association of India (NLPAI) Machine Learning Contest* (2006), 1–4.

Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014. Modelling Sarcasm in Twitter a Novel Approach. *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis* (2014), 50–58.

SK Bharti, B Vachha, RK Pradhan, KS Babu, and SK Jena. 2016. Sarcastic sentiment detection in tweets streamed in real time: a big data approach. *Digital Communications and Networks* 2, 3 (2016), 108–121.

Santosh Kumar Bharti, Korra Sathya Babu, and Sanjay Kumar Jena. 2015. 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, 1373–1380.

Aniket Dalal, Kumar Nagaraj, Uma Sawant, and Sandeep Shelke. 2006. Hindi part-of-speech tagging and chunking: A maximum entropy approach. *Proceeding of the NLPAI Machine Learning Competition* (2006), 1–4.

Chetan Dalal, Shivyansh Tandon, and Amitabha Mukerjee. 2014. Insult Detection in Hindi. (2014), 1–8.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. ACL, 107–116.

Nikita Desai and Anandkumar D Dave. 2016. Sarcasm Detection in Hindi sentences using Support Vector machine. *International Journal* 4, 7 (2016), 8–15.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of Language Resources and Evaluation Conference*. 417–422.

Navneet Garg, Vishal Goyal, and Suman Preet. 2012. Rule Based Hindi Part of Speech Tagger.. In *COLING (Demos)*. 163–174.

Roberto González-Ibánez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in Twitter: a closer look. In *Proceedings of the 49th Annual Meeting on Human Language Technologies*. ACL, 581–586.

Aditya Joshi, AR Balamurali, and Pushpak Bhattacharyya. 2010. A fall-back strategy for sentiment analysis in hindi: a case study. *Proceedings of the 8th ICON* (2010).

Nisheeth Joshi, Hemant Darbari, and Iti Mathur. 2013. HMM based POS tagger for Hindi. In *Proceeding of 2013 International Conference on Artificial Intelligence, Soft Computing (AISC-2013)*. 341–349.

Florian Kunneman, Christine Liebrecht, Margot van Mulken, and Antal van den Bosch. 2015. Signaling sarcasm: From hyperbole to hashtag. *Information Processing & Management* 51, 4 (2015), 500–509.

World Language and Culture. 2005. Top 30 Languages by Number of Native Speakers. (2005). http://www.vistawide.com/languages/top_30_languages.htm

CC Liebrecht, FA Kunneman, and APJ van den Bosch. 2013. 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, 29–37.

Peng Liu, Wei Chen, Gaoyan Ou, Tengjiao Wang, Dongqing Yang, and Kai Lei. 2014. Sarcasm Detection in Social Media Based on Imbalanced Classification. In *Web-Age Information Management*. 459–471.

Edwin Lunando and Ayu Purwarianti. 2013. Indonesian social media sentiment analysis with sarcasm detection. In *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, 195–198.

Shachi Mall and Umesh Chandra Jaiswal. 2011. Hindi Part of Speech Tagging and Translation. *Int. J. Tech* 1, 1 (2011), 29–32.

Nidhi Mishra and Amit Mishra. 2011. Part of speech tagging for Hindi corpus. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*. IEEE, 554–558.

Namita Mittal, Basant Agarwal, Garvit Chouhan, Nitin Bania, and Prateek Pareek. 2013. Sentiment analysis of hindi review based on negation and discourse relation. In *proceedings of International Joint Conference on Natural Language Processing*. 45–50.

Sneha Mulatkar. 2014. Sentiment Classification In Hindi. *International Journal of Scientific & Technology Research* 3, 5 (2014), 204–206.

Ravi Narayan, VP Singh, and S Chakraverty. 2014. Quantum neural network based parts of speech tagger for Hindi. *Int. J. Adv. Technol* 5 (2014), 137–152.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of conference on Empirical methods in natural language processing*, Vol. 10. ACL, 79–86.

Mikael Parkvall. 2007. Världens 100 största språk 2007. *The World's* 100 (2007).

Sudha Shanker Prasad, Jitendra Kumar, Dinesh Kumar Prabhakar, and Sukomal Pal. 2015. Sentiment Classification: An Approach for Indian Language Tweets Using Decision Tree. In *International Conference on Mining Intelligence and Knowledge Exploration*. Springer, 656–663.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm Detection on Twitter: A Behavioral Modeling Approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, 97–106.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In *Proceedings of the conference on empirical methods in natural language processing*. 704–714.

Delia Rusu, Lorand Dali, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. 2007. Triplet extraction from sentences. In *Proceedings of the 10th International Multiconference on Information Society-IS*. 8–12.

Yakshi Sharma, Veenu Mangat, and Mandeep Kaur. 2015. A practical approach to Sentiment Analysis of hindi tweets. In *Next Generation Computing Technologies (NGCT), 2015 1st International Conference on*. IEEE, 677–680.

Smriti Singh, Kuhoo Gupta, Manish Shrivastava, and Pushpak Bhattacharyya. 2006. Morphological richness offsets resource demand-experiences in constructing a POS tagger for Hindi. In *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, 779–786.

Justine Raju Thomas, Santosh Kumar Bharti, and Korra Sathya Babu. 2016. Automatic Keyword Extraction for Text Summarization in e-Newspapers. In *Proceedings of the International Conference on Informatics and Analytics*. ACM, 86–93.

Then, it describes the algorithms for sentiment identification in tweets and context detection in news. Finally, deployed two algorithms to detect sarcasm in Hindi tweets. Firstly, it finds a contradiction between sentiment and its context. Secondly, it identifies that tweet contradicting time-dependent fact.