Research Article

Santosh Kumar Bharti*, Reddy Naidu, and Korra Sathya Babu

Hyperbolic Feature-based Sarcasm Detection in Telugu Conversation Sentences

https://doi.org/10.1515/jisys-2018-0475
Received Jun 13, 2019; accepted Nov 03, 2019

Abstract: Recognition of sarcastic statements has been a challenge in the process of sentiment analysis. A sarcastic sentence contains only positive words conveying a negative sentiment. Therefore, it is tough for any automated machine to identify the exact sentiment of the text in the presence of sarcasm. The existing systems for sarcastic sentiment detection are limited to the text scripted in English. Nowadays, researchers have shown greater interest in low resourced languages such as Hindi, Telugu, Tamil, Arabic, Chinese, Dutch, Indonesian, etc. To analyse these low resource languages, the biggest challenge is the lack of available resources, especially in the context of Indian languages. Indian languages are very rich in morphology which pose a greater challenge for the automated machines. Telugu is one of the most popular languages after Hindi among Indian languages. In this article, we have collected and annotated a corpus of Telugu conversation sentences in the form of a question followed by a reply for sarcasm detection. Further, a set of algorithms are proposed for the analysis of sarcasm in the corpus of Telugu conversation sentences. The proposed algorithms are based on hyperbolic features namely, Interjection, Intensifier, Question mark and Exclamation symbol. The achieved accuracy is 94%.

Keywords: Hyperbolic Features, Natural Language Processing, Sarcasm, Sentiment, Telugu

2010 Mathematics Subject Classification: 68T50, 68T10, 68T05

1 Introduction

Sentiment analysis is a technique that analyses people's opinions, sentiments and emotions towards a target such as products, services, events, organizations, individuals, etc. [1]. The presence of sarcasm in the sentence makes sentiment analysis difficult as sarcasm flips the sentiment value. Therefore, sarcasm is considered as critical to identify the sentiment from a given text.

Sarcasm is a special kind of sentiment that frequently occurs during the communication between people and is mostly intentional. It is a nuanced form of language in which people state the opposite of what is implied. It can also be stated as the turbulent feature that people are often used to convey a negative meaning using only positive words or even compounded, inflated positive words [2]. An example of a simple sarcastic sentence is: "I love being ignored #sarcasm". In this example, the sentiment seems to be positive as "love" is present, but the situation is negative as "no one wants to be ignored". It means the sentence is written in a sarcastic way. It can be easily understood that sarcasm detection in the text is tough due to the lack of intonation or facial expressions. Therefore, identifying sarcasm in the text is a challenging task. Recent works in the direction of sarcasm detection have influenced local native languages. This is mainly due to the usage of

***Corresponding Author: Santosh Kumar Bharti:** Department of Computer Science and Engineering, Pandit Deendayal Petroleum University, Gandhinagar, Gujarat 382421; Email: sbharti1984@gmail.com
Reddy Naidu: Department of Computer Science and Engineering, ANITS, Sangivalasa, Visakhapatnam - 531162, India; Email: naidureddy47@gmail.com
Korra Sathya Babu: Department of Computer Science and Engineering, National Institute of Technology, Rourkela, Rourkela – 769008, India; Email: prof.ksb@gmail.com

regional languages while communicating through social media. Most of the existing algorithms for sarcasm detection are applicable for text data scripted in English [2–7]. In the domain of low resourced languages namely, Hindi, Telugu, Tamil, Arabic, Chinese, Spanish, etc., there is very little work done so far. [8–10]

In Low resourced languages domain, unavailability of the datasets is the biggest challenging task for researchers. So, it gives us a seed idea to work on sarcasm detection in this domain especially on Indian languages. Indian languages such as Hindi, Telugu, Tamil, etc. are very rich in morphology which poses another challenge for the researchers to work on it. Telugu is the second most popular language in India just after Hindi, and it has a lot of importance over other Indian languages. In the 16th century, Italian explorer Niccolo Da Conti who visited the Vijayanagara empire described Telugu as Italian of the east. Rabindranath Tagore, the well-known Bengali writer, has once heard Telugu poetry and said "Is it a language or music?", and he also said that Telugu is the sweetest of all (Indian) languages. The famous Tamil poet Subramania Bharati has sung thus "Sundara Telunginil Pattisaithu" which means "Sing in beautiful Telugu". Srikrishna Devaraya, South Indian king and a non-native speaker of Telugu said "Desabhaashalandu Telugu Lessa (Telugu is the best among all the languages in this country). Telugu being the second most spoken language in India is growing its importance and majority of the Telugu speaking social media users started communicating in their native language. An automated sentiment analyser with sarcasm detection method will enhance the better analysis of the communicated text.

In this article, we collected a corpus of Telugu conversation sentences in the form of the question followed by a reply from Telugu comedy TV shows such as "Jabardasth comedy show", "Extra Jabardasth comedy show", "Comedy Raja Band Baja", "Patas Punch", etc. This corpus comprises of conversation between different comedy actors. It is mostly in the form of a question followed by a sarcastic reply. These replies are considered as sarcastic in the context of the question. Three algorithms have been devised by analysing the corpus for sarcasm detection. These three proposed algorithms are devised based on the occurrences of hyperbolic features in the Telugu conversation sentences.

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

2 Related Work

This section gives a survey on existing methods for sarcasm detection. Majority of the work in sarcasm detection has been done in English language as it is the most dominating language used in social media for communication. In recent times, sarcasm detection on English scripted domains such as Twitter data, product reviews, website comments, etc., were done tremendously by many researchers [2–5, 7, 11]. In the domain of Low resourced languages such as Hindi, Telugu, Tamil, Chinese, Arabic, etc., very little work has been done [8–10]. The following subsections will detail about sarcasm detection in English and low resourced languages.

2.1 Sarcasm Detection on English Language

Lexical features play a vital role in detecting irony and sarcasm in text [12]. Lexical features along with syntactic features were used to detect sarcastic tweets. A semi-supervised approach [11] was used to detect sarcasm in tweets and Amazon product reviews. They used two interesting lexical features, namely pattern-based (high-frequency words and content words) and punctuation-based to build a weighted K-Nearest Neighbor (KNN) classification model to perform sarcasm detection. Numerous lexical features derived from linguistic inquiry and word count [13], WordNet affect [14] and pragmatic features such as emoticons, smiles and replies were explored [3] to identify sarcasm in tweets. A well-constructed lexicon-based approach was used to detect sarcasm based on an assumption that sarcastic tweets are a contrast between a positive sentiment and a negative situation [5] and for lexicon generation, they used unigram, bigram and trigram features. The Intensifier is used as hyperbole features to detect sarcasm in tweets as utterance with a hyperbole. For example - 'fantastic weather when it rains' is identified as sarcastic with more ease than the utterance without a hyperbole like - 'the weather is good when it rains' [4]. The utterance with the hyperbole 'fantastic' may be easier to interpret more sarcastic than the utterance with the non-hyperbolic 'good'. Interjection words are used as hyperbole feature to identify sarcasm in tweets [2]. They also used a parsing technique to divide a tweet into phrases to generate the lexicon file to identify sarcasm in Twitter data. Rather than lexical and linguistic traits, the Twitter user's behavioral trait is used as the feature for sarcasm detection. Behavioral context was used to convey the sarcasm and employed theories from behavioral and psychological studies to construct a behavioral modeling framework tuned for detecting sarcasm [6]. Sarcasm requires some shared knowledge between speaker and audience, and it is a profoundly contextual phenomenon. Most computational approaches to sarcasm detection, however, treat it as a purely linguistic matter, using information such as lexical cues and their corresponding sentiment as predictive features [15]. A system was develpoed that identifies sarcastic tweets to predict the result of an upcoming election by analysing people's opinion on Twitter [16]. They used Exclamation mark (!), Question mark (?), Hashtag sarcasm and Irony, Emoticons, Adjectives and Verbs as features to identify sarcastic polarity in Twitter data using supervised machine learning approach. The author's past tweets can provide an additional context for sarcasm detection. They exploited the author's past sentiment on the entities in a tweet to detect the sarcastic intent [17]. A framework was introduced based on the linguistic theory of context incongruity and inter-sentential incongruity for sarcasm detection by considering the previous post in the discussion thread [18]. A Hadoop based framework that captures a massive amount of real-time tweets and processes it with a set of algorithms that identify sarcastic sentiment efficiently was also proposed [7].

2.2 Sarcasm Detection on Low-Resourced Languages

The first work on detecting sarcasm on Low-Resourced languages was done in Indonesian social media data [8]. The dataset was gathered manually from Twitter and proposed two additional features to detect sarcasm after a common sentiment analysis was conducted. The features are the negativity information and the number of Interjection words. They also employed translated SentiWordNet in the sentiment classification. Thelwall *et al.* [19] have provided a huge number of informal messages posted every day on social network sites, blogs and discussion forums. Till date algorithms are devised to identify sentiment and sentiment strength that help to understand the role of emotion in this informal communication and also to identify inappropriate or anomalous affective utterances, potentially associated with threatening behaviour to the self or others. A set of features specifically for detecting sarcasm in social media were introduced and had deployed a novel Multi Strategy Ensemble Learning Approach (MSELA) to handle imbalance problem in English and Chinese sentences [9]. A system was proposed to detect sarcastic sentences in Hindi language using Support Vector Machines [10]. They focused on features like Emoticons and Punctuation marks for sarcasm detection. As per our best knowledge, no work on sarcasm detection in Telugu is found so far.

3 Proposed Scheme

This section describes the model for sarcasm detection followed by the process of Telugu data collection and annotation. It also explains the POS tagging followed by tagged data analysis to form the rules for sarcasm detection in Telugu conversation sentences.

3.1 Model for Sarcasm Detection

The pipeline process of sarcasm detection in Telugu conversation sentences is shown in Figure 1. It starts with data collection followed by data annotation. In the next step, we identify the appropriate POS tag information of the annotated Telugu sentences. Further, each tagged data was analysed for categorization of sentences from the occurrences of hyperbole features namely, Interjection, Intensifier, Question mark and Exclamation mark. Based on these hyperbole features, a set of algorithms are proposed for sarcasm detection in each category.

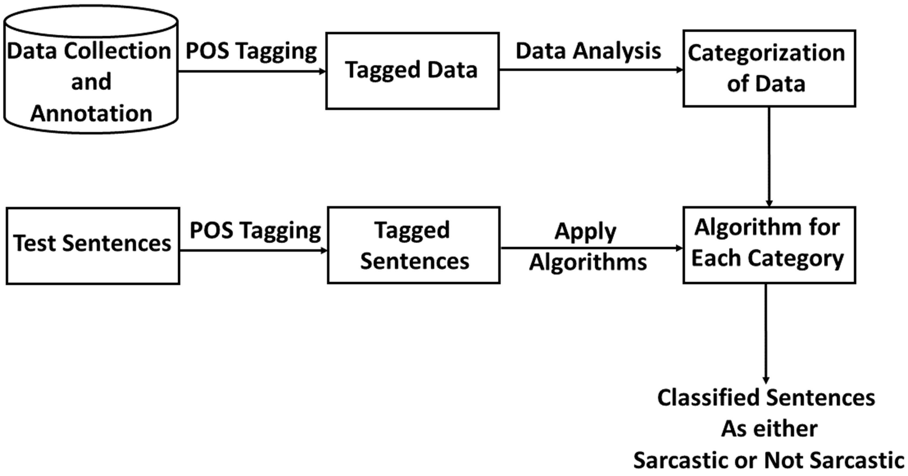

Data Collection and Annotation
POS Tagging
Tagged Data
Data Analysis
Categorization of Data
Test Sentences
POS Tagging
Tagged Sentences
Apply Algorithms
Algorithm for Each Category
Classified Sentences As either Sarcastic or Not Sarcastic

Figure 1: Model for Sarcasm Detection

Further, for testing process, the test sentences are initially fed to the process of POS tagger to obtain the tagged sentences. Now, these tagged sentences are applied on the algorithms to classify as either sarcastic or not sarcastic.

3.2 Data Collection

Since, Telugu is a low resourced language, the availability of the data is very rare on the Internet. So, we have collected it manually from different sources such as TV series, Web, Internet, etc. The process of data

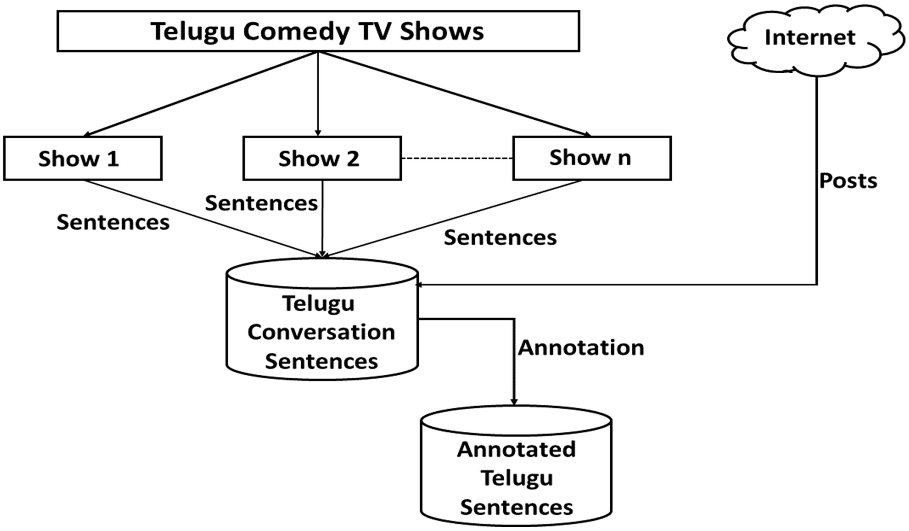


Figure 2: Manual Process for Data Collection from Telugu TV Shows

collection and annotation are shown in Figure 2. We have collected around 5500 Telugu sentences. Majority of the conversation sentences are from various Telugu TV comedy series of the ETV plus channel namely, “Jabardasth”, “Extra Jabardasth”, “Pataas”, “Cinema Chupista Mava”, “Express Raja”, “Naa Show Naa Istam” etc. To collect these sentences, nearly 350 archive episodes were watched all together. As the sentences were taken from video, collected data are in the form of conversation sentences between two or more. Therefore, the structure of the sentences is in the form of the question followed by a reply. The collected dataset has been made publicly available through the GitHub. One can find data on the link https://github.com/sbharti1984/Telugu-Sarcastic-Sentences.

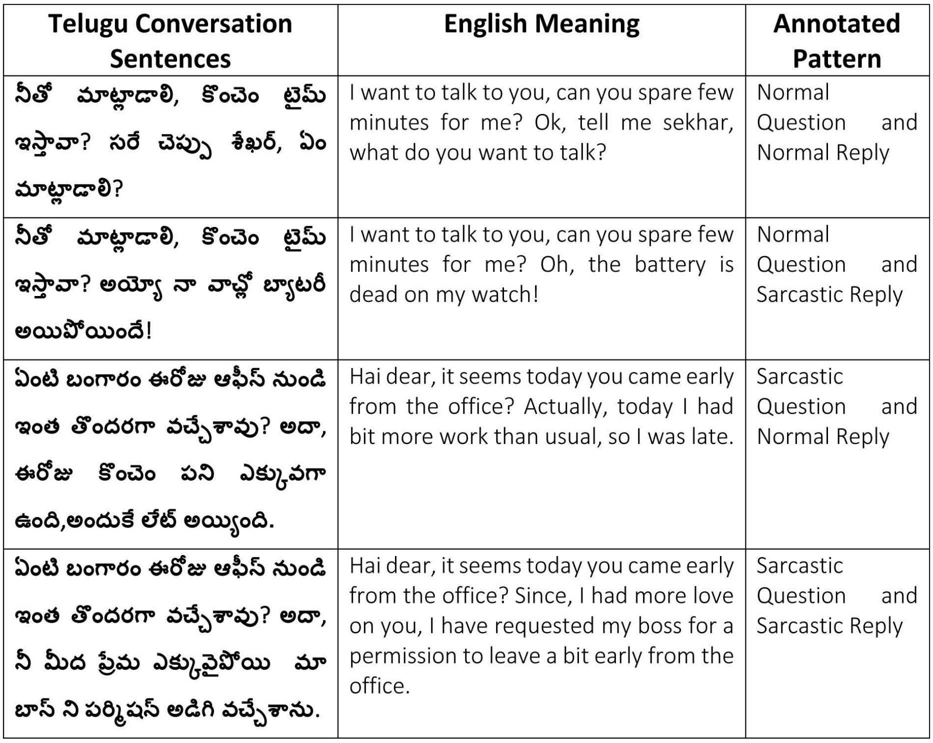
3.3 Data Annotation

The collected dataset was distributed among professionals in the Telugu language who are teachers and practitioners. They gave very good response and annotated these 5500 sentences manually to find the sentence as sarcastic or not. After collecting the annotation results from all the three individuals, we observe that the structure of conversation sentences followed any one of the following patterns:

1. Normal question followed by a Normal reply.
2. Normal question followed by a Sarcastic reply.
3. Sarcastic question followed by a Normal reply.
4. Sarcastic question followed by a Sarcastic reply.

A list of sample annotated conversation sentences (one for each above patterns) that are collected is shown in Figure 3. Based on annotation, we classified the sentences of the dataset and observed that most of the sentences belong to the second pattern: “normal question followed by a sarcastic reply”.

To measure the Inter-Annotator Agreement (IAA), there are two coefficients such as Cohen’s Kappa [20] and Fleiss Kappa [21]. If the annotators are two or more than two, The Cohen’s Kappa or Fleiss Kappa is used



Telugu Conversation Sentences	English Meaning	Annotated Pattern
నీతో మాట్లాడాలి, కొంచెం టైమ్ ఇస్తావా? సరే చెప్పు శేఖర్, ఏం మాట్లాడాలి?	I want to talk to you, can you spare few minutes for me? Ok, tell me sekhar, what do you want to talk?	Normal Question and Normal Reply
నీతో మాట్లాడాలి, కొంచెం టైమ్ ఇస్తావా? అయ్యో నా వాచ్లో బ్యాటరీ అయిపోయిందే!	I want to talk to you, can you spare few minutes for me? Oh, the battery is dead on my watch!	Normal Question and Sarcastic Reply
ఏంటి బంగారం ఈరోజు ఆఫీస్ నుండి ఇంత తొందరగా వచ్చేశావు? అదా, ఈరోజు కొంచెం పని ఎక్కువగా ఉంది,అందుకే లేట్ అయ్యింది.	Hai dear, it seems today you came early from the office? Actually, today I had bit more work than usual, so I was late.	Sarcastic Question and Normal Reply
ఏంటి బంగారం ఈరోజు ఆఫీస్ నుండి ఇంత తొందరగా వచ్చేశావు? అదా, నీ మీద ప్రేమ ఎక్కువైపోయి మా బాస్ ని పర్మిషన్ అడిగి వచ్చేశాను.	Hai dear, it seems today you came early from the office? Since, I had more love on you, I have requested my boss for a permission to leave a bit early from the office.	Sarcastic Question and Sarcastic Reply

Figure 3: Sample of Annotated Telugu Conversation Sentences

respectively.
In this work, as the annotators are three (more than two), we approached Fleiss Kappa coefficient, and the formula for that is shown in Equation 1. We got the IAA as 0.85, which is said to be perfect agreement.

$$k = \frac{\overline{P} - \overline{P_e}}{1 - \overline{P_e}} \tag{1}$$

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

In this article, our assumptions are as follows:

1. The sentences belonging to the pattern “normal question followed by normal reply” are non-sarcastic sentences.
2. The sentences belonging to the pattern “normal question followed by sarcastic reply” are sarcastic sentences.
3. The sentences belonging to the pattern “sarcastic question followed by normal reply” are sarcastic, but the frequency of occurrences is very rare.
4. The sentences belonging to the pattern “sarcastic question followed by sarcastic reply” are sarcastic, but the frequency of occurrences is very rare.

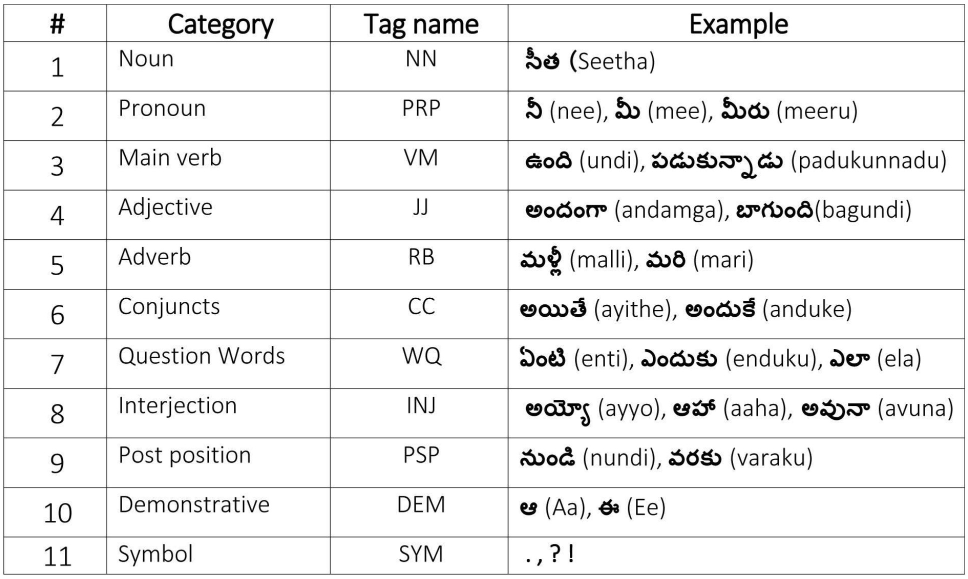
With these assumptions, we observed that out of 5500 sentences, 5200 sentences were sarcastically annotated and rest 300 was not sarcastic. In this work, we considered only sentences that follow “normal question followed by sarcastic reply” pattern as a sarcastic sentences. The occurrences of this pattern are very frequent and comprise of approximately 97% of the sarcastic sentences dataset. The sentences belonging to the patterns “sarcastic question followed by normal reply” and “sarcastic question followed by sarcastic reply” are omitted because of the rarity. Based on this assumption, we have developed the rules to detect the sarcasm in the given sentence. For this experiment, the analysis dataset, training and testing set are as follows:

1. After annotation, 5200 sentences were found sarcastic in a total of around 5500 sentences.
2. 5044 (97% of 5200) sarcastic sentences followed the pattern normal question followed by sarcastic reply.
3. All 5044 sarcastic sentences were used for the analysis to develop the rules which will detect the sarcasm.
4. 188 (94 sarcastic and 94 non-sarcastic) sentences were used as testing set which are not part of the dataset.

3.4 POS Tagging

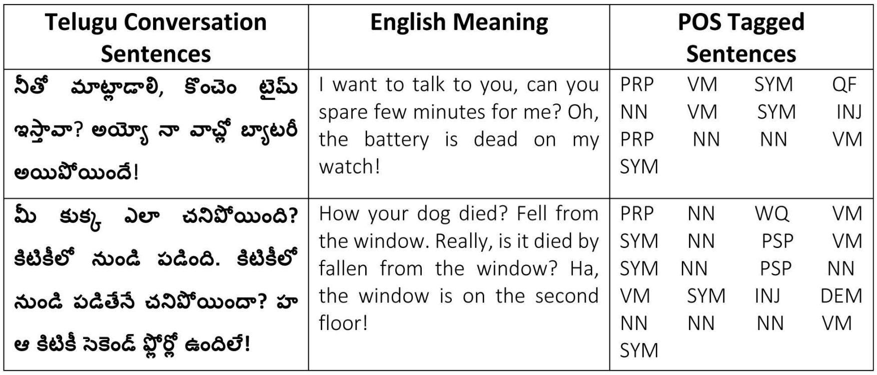
POS tagging is the process of assigning a correct POS tag such as Noun, Verb, Adverb, etc., to each word of the given input sentence. POS taggers are developed by modelling the morphosyntactic structure of NLP . The Telugu tagger is similar to the model 5 described in Table 2 of [22], but with a focus on Telugu. The corpora are downloaded, cleaned and tagged with a high Precision and low Recall tagger. As the tagger is trained on large data, the tagger is expected to handle large vocabulary and also predicting the tags of unknown words using known words. They followed HMM-based approach and the Indian language standard tagset [23] which comprise 21 tags to build the tagger. The available Telugu tagger is based on TnT tagger, which is well known for its robustness and speed.

Some of the Telugu tags used in this article are shown in Figure 4. An example of Telugu sentences with corresponding POS tag information is shown in Figure 5.



#	Category	Tag name	Example
1	Noun	NN	సీత (Seetha)
2	Pronoun	PRP	నీ (nee), మీ (mee), మీరు (meeru)
3	Main verb	VM	ఉంది (undi), పడుకున్నాడు (padukunnadu)
4	Adjective	JJ	అందంగా (andamga), బాగుంది(bagundi)
5	Adverb	RB	మళ్లీ (malli), మరి (mari)
6	Conjuncts	CC	అయితే (ayithe), అందుకే (anduke)
7	Question Words	WQ	ఏంటి (enti), ఎందుకు (enduku), ఎలా (ela)
8	Interjection	INJ	అయ్యో (ayyo), ఆహ్ (aaha), అవునా (avuna)
9	Post position	PSP	నుండి (nundi), వరకు (varaku)
10	Demonstrative	DEM	ఆ (Aa), ఈ (Ee)
11	Symbol	SYM	. , ? !

Figure 4: List of POS tagset used in this work



Telugu Conversation Sentences	English Meaning	POS Tagged Sentences
నీతో మాట్లాడాలి, కొంచెం టైమ్ ఇస్తావా? అయ్యో నా వాచ్లో బ్యాటరీ అయిపోయిందే!	I want to talk to you, can you spare few minutes for me? Oh, the battery is dead on my watch!	PRP VM SYM QF NN VM SYM INJ PRP NN NN VM SYM
మీ కుక్క ఎలా చనిపోయింది? కిటికీలో నుండి పడింది. కిటికీలో నుండి పడితేనే చనిపోయిందా? హ ఆ కిటికీ సెకెండ్ ఫ్లోర్లో ఉందిలే!	How your dog died? Fell from the window. Really, is it died by fallen from the window? Ha, the window is on the second floor!	PRP NN WQ VM SYM NN PSP VM SYM NN PSP NN VM SYM INJ DEM NN NN NN VM SYM

Figure 5: Example of POS tagged data in Telugu sentences

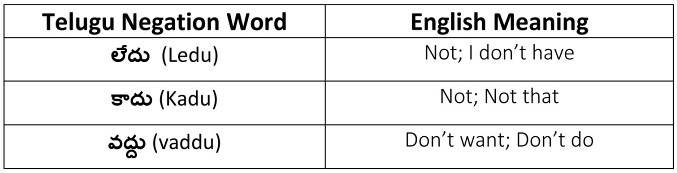
3.5 Proposed Algorithms

In this article, we analysed 4950 sarcastic sentences for training set and observed that these sentences could be classified into three categories:

1. Sentence that start with Telugu negation word.
2. Sentence that start with Telugu interjection word.
3. Reply of a sentence in the form of a question.

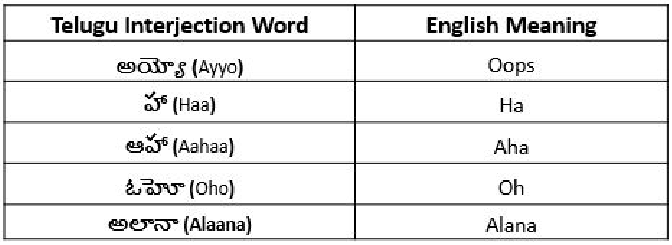
A list of Telugu negation words and Telugu interjection words are given in Figures 6 and 7 respectively. Based on the observation from Telugu conversation sarcastic sentences, a set of three algorithms are proposed to detect sarcasm in each of the categories. The proposed algorithms are as follows:

1. $Telugu\_Negation\_Word\_Start(TNWS)$
2. $Telugu\_Interjection\_Word\_Start(TIWS)$
3. $Reply\_in\_Form\_of\_Question(RiFoQ)$



Telugu Negation Word	English Meaning
లేదు (Ledu)	Not; I don’t have
కాదు (Kadu)	Not; Not that
వద్దు (vaddu)	Don’t want; Don’t do

Figure 6: List of Telugu Negation Words



Telugu Interjection Word	English Meaning
అయ్యో (Ayyo)	Oops
హా (Haa)	Ha
ఆహా (Aahaa)	Aha
ఓహో (Oho)	Oh
అలానా (Alaana)	Alana

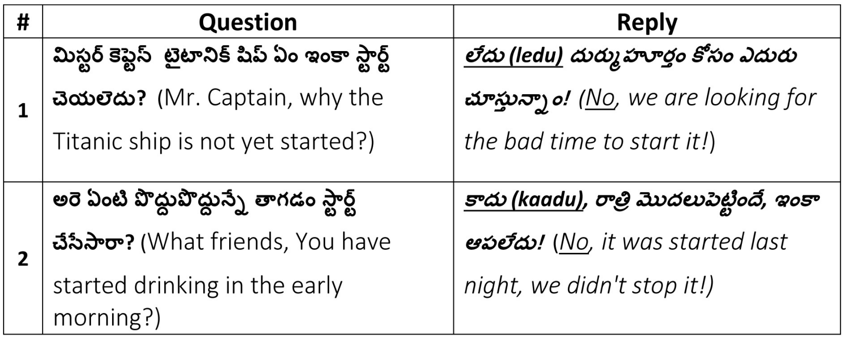
Figure 7: List of Telugu Interjection Words

In this article, we proposed three algorithms for identifying sarcasm in Telugu conversation sentences as given in algorithms 1, 2, and 3. Here, we explained the working procedure of all the three algorithms using examples.

3.5.1 *Telugu_Negation_Word_Start*(*TNWS*)

This algorithm is based on Telugu negation words *i.e.* “ledu”, “kaadu” and “vaddu”. During the analysis of sarcastic sentences, we observed that these negation words frequently appear as a starting word in sarcastic reply of the conversation sentences as shown in Figure 8. Based on this observation, we proposed an algorithm for the sentences whose reply starts with Telugu negation word as shown in Algorithm 1.

Algorithm 1 takes the corpus of Telugu conversation sentences (C) as an input and extracts the first word and last word of every sentence and store it in F_tok and L_tok respectively. Next, it compares F_tok with Telugu negation words, *i.e.*, “ledu”, “kaadu”, “vaddu” and L_tok with a question mark (?) and exclamation mark (!). If F_tok match with any of the above mentioned Telugu Negation words and L_tok match with either Exclamation or Question mark, then the given sentence is classified as sarcastic, otherwise, check for other two proposed algorithms. Next, we find POS tag information of F_tok and store it in FT. If FT is the Telugu interjection word as shown in Figure 7, then those sentences are fed into Algorithm 2 and rest of the sentences are fed into Algorithm 3.



#	Question	Reply
1	**మిస్టర్ కెప్టెన్ టైటానిక్ షిప్ ఏం ఇంకా స్టార్ట్ చెయలెదు?** (Mr. Captain, why the Titanic ship is not yet started?)	***లేదు (ledu) దుర్ముహూర్తం కోసం ఎదురు చూస్తున్నాం!*** (*No, we are looking for the bad time to start it!*)
2	**అరె ఏంటి పొద్దుపొద్దున్నే తాగడం స్టార్ట్ చేసేసారా?** (What friends, You have started drinking in the early morning?)	***కాదు (kaadu), రాత్రి మొదలుపెట్టిందే, ఇంకా ఆపలేదు!*** (*No, it was started last night, we didn't stop it!)*

Figure 8: Telugu conversation sentences for Algorithm 1

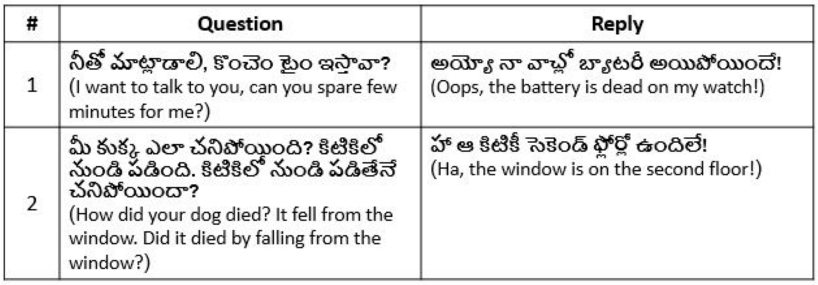
Algorithm 1: *Telugu_Negation_Word_Start*(*TNWS*)

```
Input: Corpus of Telugu Conversation Sentences (C)
Output: Classified sentences into either Sarcastic or not Sarcastic
Notation: S: Sentence, C: Corpus, tok: Token
while S in C do
    F_tok = find_first_tok(S)
    L_tok = find_last_tok(S)
    if (F_tok == 'ledu' | 'kadu' | 'vaddu') && (L_tok == '!'|'?') then
        S is classified as Sarcastic.
    end
    else
        FT = find_POS_tag(F_tok)
        if (FT == 'INJ') then
            Apply Telugu_Interjection_Word_Start (S) algorithm for Sarcasm Detection
        end
        else
            Apply Reply_in_Form_of_Question (S) algorithm for Sarcasm Detection
        end
    end
end
```

Algorithm 1 is based on Telugu negation word as examples shown in Figure 8. According to Algorithm 1, any reply starts with one of the negation word given in Table 6 and ends with either (?) or (!) then sentence will be classified as sarcastic. In Figure 8, the given Telugu sentences are in the form of a question followed by a reply. The reply of the first sentence starts with Telugu negation word "ledu" and ends with exclamation symbol (!). Therefore, the given sentence is sarcastic. Similarly, the reply of the second sentence is starts with negation word "kaadu" and ends with exclamation symbol (!). Therefore, the given sentence is sarcastic. In proposed Algorithm 1, we have considered those negation words that occur very frequently in sarcastic Telugu conversation sentences. These negation words act as an intensifier in reply.

3.5.2 *Telugu_Interjection_Word_Start*(*TIWS*)

This algorithm is based on Telugu interjection words such as "ayyo", "haa", "alaana", etc. While analysing Telugu sarcastic sentences, we observed that many sarcastic replies start with Telugu interjection words as shown in Figure 9. Based on this observation, we proposed an algorithm for the sentences whose reply starts with Telugu interjection words as shown in Algorithm 2.



#	Question	Reply
1	నీతో మాట్లాడాలి, కొంచెం టైం ఇస్తావా? (I want to talk to you, can you spare few minutes for me?)	అయ్యో నా వాచ్లో బ్యాటరీ అయిపోయిందే! (Oops, the battery is dead on my watch!)
2	మీ కుక్క ఎలా చనిపోయింది? కిటికిలో నుండి పడింది. కిటికిలో నుండి పడితేనే చనిపోయిందా? (How did your dog died? It fell from the window. Did it died by falling from the window?)	హా ఆ కిటికీ సెకెండ్ ఫ్లోర్లో ఉందిలే! (Ha, the window is on the second floor!)

Figure 9: Telugu conversation sentences for Algorithm 2

```
Algorithm 2: Telugu_Interjection_Word_Start(TIWS)
  Input: Telugu Conversation Sentence (S)
  Output: Classified into either Sarcastic or not Sarcastic.
  Notation: $S$: Sentence, $VM$: Main Verb, $tok$: Token, $JJ$: Adjective, $INJ$: Interjection, $PRP$: Pronoun,
    $NN$: Noun, $TF$: Tag File, $ANT$: Any Next Tag
  $TF \leftarrow find\_pos\_tag\ (S)$
  $FT = find\_first\_tag(TF)$
  $ST = find\_second\_tag(TF)$
  $TT = find\_third\_tag(TF)$
  while $tag\ in\ TF$ do
      if $(FT == `INJ') \&\& (ANT == (`NN' + `VM')$ then
          S is classified as Sarcastic.
      end
      else if $(FT == `INJ') \&\& (ST == `JJ')$ then
          S is classified as Sarcastic.
      end
      else if $(FT == `INJ') \&\& (ST == `PRP') \&\& (TT == `NN' | `VM')$ then
          S is classified as Sarcastic.
      end
      else
          S is classified as not sarcastic
      end
  end
```

Algorithm 2 takes a sentence as input that start with interjection word and finds the POS tag information for every sentence and append it to file TF. Next, it finds the first, second and the third tag and stores in FT, ST and TT respectively. If FT is an Interjection (INJ) and ST is an Adjective (JJ), then the sentence is classified as sarcastic. Otherwise, if the FT is an INJ and a bigram tag (NN + VM) sequence is present anywhere in the rest of sentence, the sentence is classified as sarcastic. Finally, if FT is INJ and ST is a pronoun (PRP), and TT is either main verb (VM) or noun (NN), then the sentence is classified as sarcastic. Otherwise, the sentence is not sarcastic.

Algorithm 2 is based on Telugu interjection word as examples shown in Figure 9. According to Algorithm 2, any reply starts with one of the interjection word given in Table 7 and POS tag value either a noun or verb present anywhere in remaining part then sentence will be classified as sarcastic. Similarly, other rules are given. In Figure 9, the given Telugu sentences are in the form of a question followed by a reply. The POS tags sequence for the reply of the first sentence is: “INJ SYM PRP NN NN VM SYM”. The reply of the first sentence starts with Telugu interjection tag “INJ”, and noun (NN) appears at $4^{th} and 5^{th}$ position or verb (VM) appears at 6^{th} position. Therefore, given sentence is classified as sarcastic. The one condition is sufficient either presence of NN or VM. Similarly, for the second sentence as well. In proposed Algorithm 2, we have considered those interjection words that occur very frequently in sarcastic Telugu conversation sentences.

3.5.3 *Reply_in_Form_of_Question(RiFoQ)*

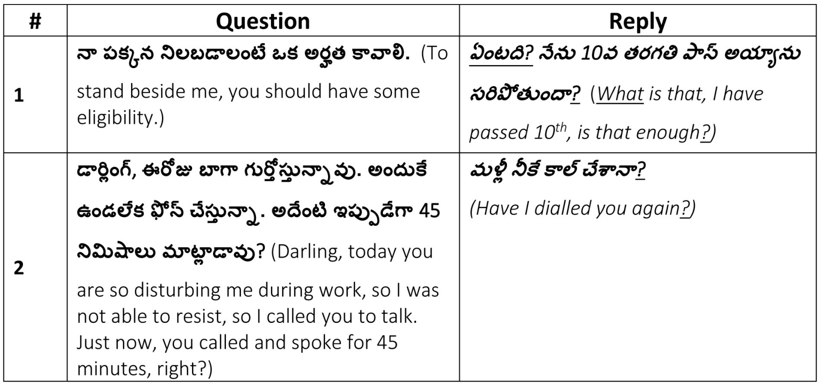
We observed that several sarcastic replies were in the form of a question during analysis of Telugu sarcastic sentences as shown in Figure 10. Therefore, we proposed an algorithm for the conversation sentences whose reply was in the form of a question and shown in Algorithm 3.

Algorithm 3 takes rest of the sentences as an input that neither starts with Telugu Negation word nor Telugu interjection words. Next, it finds the POS tag information for every sentence and appends it to file TF.

Algorithm 3: *Reply_in_Form_of_Question*(*RIFOQ*)

```
Input: Telugu conversation sentence (S).
Output: Classified into either Sarcastic or not Sarcastic.
Notation: $S$: Sentence, $TF$: Tag File, $VM$: Main Verb, $tok$: Token, $RB$: Adverb, $WQ$: Question Words,
  $PRP$: Pronoun, $NN$: Noun
$TF \leftarrow find\_pos\_tag\ (S)$
$FT = find\_first\_tag\ (TF)$
$ST = find\_second\_tag\ (TF)$
$TT = find\_third\_tag\ (TF)$
$SLT = find\_second\_last\_tag\ (TF)$
$LT = find\_last\_tok\ (TF)$
while $tag\ in\ TF$ do
    if $(tag == `WQ')\ \&\&\ (SLT == (`VM'|`NN'))\ \&\&\ (LT == `?')$ then
        S is classified as Sarcastic.
    end
    else if $(FT == `VM')\ \&\&\ (ST == `NN'|`VM')\ \&\&\ (SLT == (`VM'|`NN'))\ \&\&\ (LT == `?')$ then
        S is classified as Sarcastic.
    end
    else if $(FT == `RB')\ \&\&\ (ST == `PRP')\&\&\ (TT == `NN')\ \&\&\ (SLT == (`VM'|`NN'))\&\&\ (LT == `?')$
      then
        S is classified as Sarcastic.
    end
    else
        S is classified as not Sarcastic
    end
end
```

In the next step, it finds the first, second, third and the second-last-tag of every sentence and stores it in FT, ST, TT and SLT respectively. If 'WQ' tag is present in TF and in the corresponding sentence, if SLT is either NN or VM and the LT is '?', then the sentence is classified as sarcastic. Otherwise, if FT is VM, ST and SLT is either VM or NN and the LT is '?', then the sentence is classified as sarcastic. Finally, if FT is RB, ST is PRP, TT is NN, SLT is either NN or VM and the LT is '?', then the sentence is classified as sarcastic. Otherwise, the sentence is not sarcastic.



#	Question	Reply
1	**నా పక్కన నిలబడాలంటే ఒక అర్హత కావాలి.** (To stand beside me, you should have some eligibility.)	*ఏంటది? నేను 10వ తరగతి పాస్ అయ్యాను సరిపోతుందా? (What is that, I have passed 10th, is that enough?)*
2	**డార్లింగ్, ఈరోజు బాగా గుర్తొస్తున్నావు. అందుకే ఉండలేక ఫోన్ చేస్తున్నా. అదేంటి ఇప్పుడేగా 45 నిమిషాలు మాట్లాడావు?** (Darling, today you are so disturbing me during work, so I was not able to resist, so I called you to talk. Just now, you called and spoke for 45 minutes, right?)	*మళ్లీ నీకే కాల్ చేశానా? (Have I dialled you again?)*

Figure 10: Telugu conversation sentences for Algorithm 3

Algorithm 3 is based on a unique feature as a reply of a conversation sentence is in the form of a question. To identify a reply is in the form of question, one need to check the presence of question mark tag (WQ) and end symbol is question mark (?). Some other rules are given in Algorithm 3. In Figure 10, the given Telugu sentences are in the form of a question followed by a reply. The POS tags sequence for the reply of the first sentence is: "WQ SYM PRP QC NN NN NN VM VM SYM". According to the algorithm, reply contains a question tag (WQ), and last two tags are either a verb (VB) followed by symbol or noun (NN) followed by a symbol. The symbol is a question mark (?). Therefore, given sentence is classified as sarcastic. Similarly, the POS tags sequence for second sentence‘s reply is: "RB PRP NN VM SYM". According to the other rule, if any reply starts with tag adverb (RB) followed by the second tag is a pronoun (PRP) followed by the third tag is a noun (NN), and last two tags are a verb (VM) and question mark symbol (?) then given sentence is classified as sarcastic.

4 Results

This section describes the performance of the proposed algorithms to detect sarcasm in Telugu conversation sentences.

4.1 Statistical Evaluation Metrics

There are three statistical parameters namely, $Precision$, $Recall$ and $F-score$ used to evaluate the proposed approaches. $Precision$ shows how much relevant information is identified correctly and $Recall$ shows how much extracted information is relevant. $F-score$ is the harmonic mean of $Precision$ and $Recall$. Equations 2, 3, and 4 shows the formula to calculate $Precision$, $Recall$ and $F-score$.

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

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

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

where,
T_p = True Positive, F_p = False Positive, F_n = False Negative.

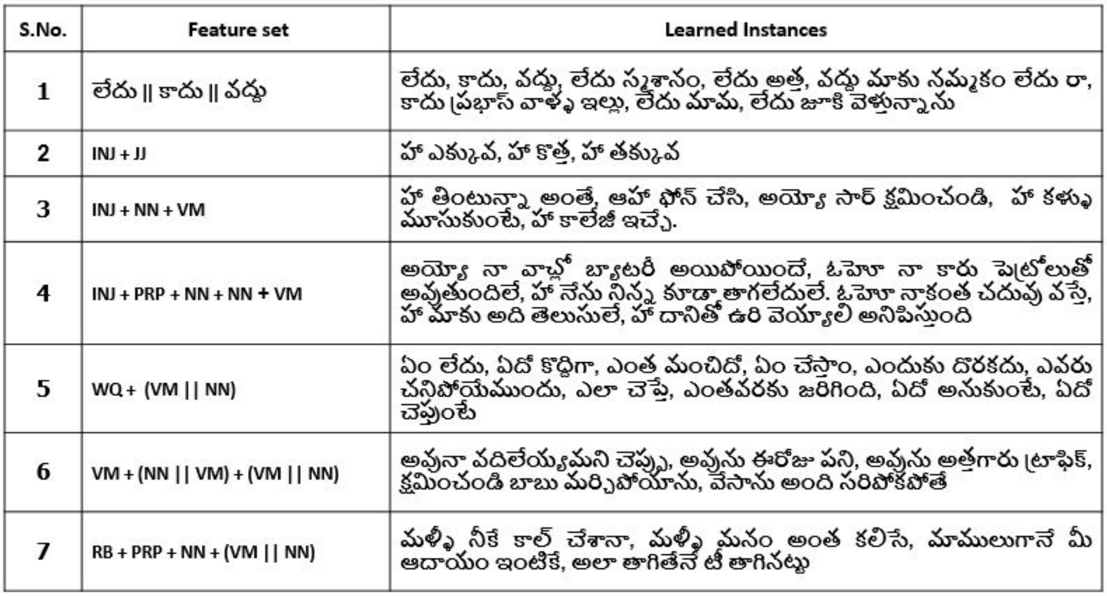
4.2 Analysis with Machine Learning Approaches

In this article, we performed analysis of proposed algorithms using various machine learning approaches as shown in Table 5. The features are extracted by analysing the annotated sentences. The Figure 11 shows the features used to train the classifiers and sample of learned instances of each feature.

The experiment was done on all the proposed algorithms individually and combined of all three algorithms by performing 5 trials as well as varying the training and testing split ratio as start with 20 upto 100. The brief introduction of all the used classifiers in this work are as follows:

The performance of Algorithm 1 is shown in Figure 12. It is observed that, the highest accuracy of 90% is reported with a train_test split ratio of (80-20) by the all classifiers. After that, for any train_test split ratio, all classifiers reports the same except the Naive Bayes which fluctuates over the variations of ratio. All the 5 trials of (80-20) split by the classifiers is shown in Table 1.

The Figure 13 shows the performance of Algorithm 2. It is observed that, the maximum accuracy of 86.43% is reported with a train_test split ratio of (60-40) by all the classifiers except the Naive Bayes. All the 5 trials of (60-40) split by the classifiers is shown in Table 2.



S.No.	Feature set	Learned Instances
1	లేదు || కాదు || వద్దు	లేదు, కాదు, వద్దు, లేదు స్మశానం, లేదు అత్త, వద్దు మాకు నమ్మకం లేదు రా, కాదు ప్రభాస్ వాళ్ళ ఇల్లు, లేదు మామ, లేదు జూకి వెళ్తున్నాను
2	INJ + JJ	హా ఎక్కువ, హా కొత్త, హా తక్కువ
3	INJ + NN + VM	హా తింటున్నా అంతే, ఆహా ఫోన్ చేసి, అయ్యో సార్ క్షమించండి, హా కళ్ళు మూసుకుంటే, హా కాలేజీ ఇచ్చే.
4	INJ + PRP + NN + NN + VM	అయ్యో నా వాచ్లో బ్యాటరీ అయిపోయిందే, ఓహో నా కారు పెట్రోలుతో అవుతుందిలే, హా నేను నిన్న కూడా తాగలేదులే. ఓహో నాకంత చదువు వస్తే, హా మాకు అది తెలుసులే, హా దానితో ఉరి వెయ్యాలి అనిపిస్తుంది
5	WQ + (VM || NN)	ఏం లేదు, ఏదో కొద్దిగా, ఎంత మంచిదో, ఏం చేస్తాం, ఎందుకు దొరకదు, ఎవరు చనిపోయేముందు, ఎలా చెప్తే, ఎంతవరకు జరిగింది, ఏదో అనుకుంటే, ఏదో చెప్తుంటే
6	VM + (NN || VM) + (VM || NN)	అవునా వదిలేయ్యమని చెప్పు, అవును ఈరోజు పని, అవును అత్తగారు ట్రాఫిక్, క్షమించండి బాబు మర్చిపోయాను, వేసాను అంది సరిపోకపోతే
7	RB + PRP + NN + (VM || NN)	మళ్ళీ నీకే కాల్ చేశానా, మళ్ళీ మనం అంత కలిసే, మాములుగానే మీ ఆదాయం ఇంటికే, అలా తాగితేనే టీ తాగినట్టు

Figure 11: Decision Tree Classifier Algorithm.

Table 1: 5 trials of the classifiers with a (80-20) split

Trial	NB	SVM	DT	KNN	RF	AB
1	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**	**0.9**
2	0.85	0.85	0.85	0.85	0.85	0.85
3	0.875	0.875	0.875	0.875	0.875	0.875
4	0.75	0.75	0.75	0.75	0.75	0.75
5	0.775	0.775	0.775	0.775	0.775	0.775

Table 2: 5 trials of the classifiers with a (60-40) split

Trial	NB	SVM	DT	KNN	RF	AB
1	0.7037	0.8148	0.8148	0.8148	0.8148	0.2469
2	0.6543	0.7530	0.7530	0.7530	0.7530	0.7530
3	0.2469	0.7530	0.7530	0.7530	0.7530	0.7530
4	0.7777	0.7777	0.7777	0.7777	0.7777	0.7283
5	**0.1728**	**0.8641**	**0.8641**	**0.8641**	**0.8641**	**0.8641**

Table 3: 5 trials of the classifiers with a (60-40) split

Trial	NB	SVM	DT	KNN	RF	AB
1	0.7777	0.7777	0.7777	0.7777	0.7777	0.5802
2	**0.8765**	**0.8765**	**0.876**5	**0.8765**	**0.8765**	**0.8765**
3	0.7901	0.7901	0.7901	0.7901	0.7901	0.7901
4	0.8395	0.8395	0.8395	0.8395	0.8395	0.8395
5	0.8271	0.8271	0.8271	0.8271	0.8271	0.8271

The Figure 14 depicts the performance of Algorithm 3. It is observed that, the maximum accuracy of 87.65% is reported with a train_test split ratio of (60-40) by all the classifiers. All the 5 trials of (60-40) split by the classifiers is shown in Table 3.

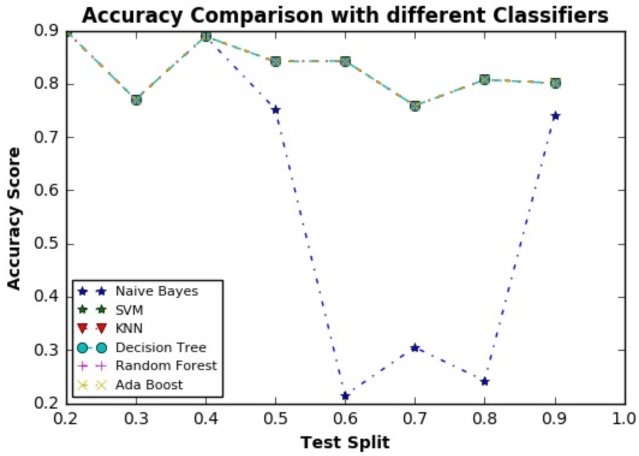


Figure 12: Accuracy of Algorithm 1 of Different Classifiers

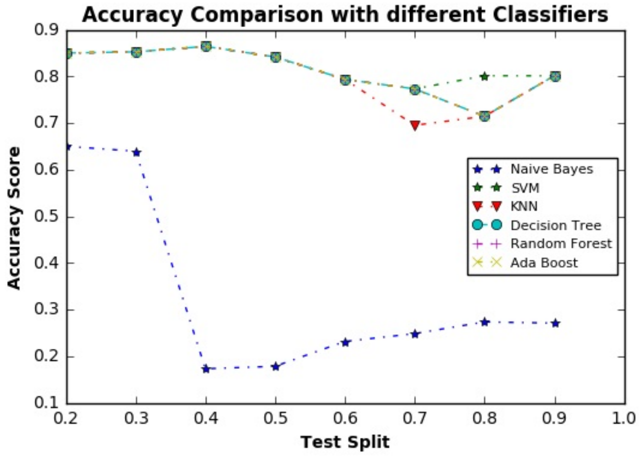


Figure 13: Accuracy of Algorithm 2 of Different Classifiers

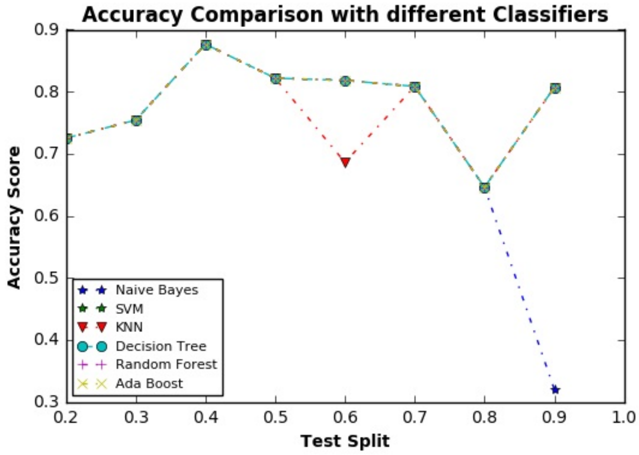


Figure 14: Accuracy of Algorithm 3 of Different Classifiers

The performance of the combined Algorithm is depicted through Figure 15. It is observed that, the maximum accuracy of 85% is reported with a train_test split ratio of (80-20) by all the classifiers. All the 5 trials of (80-20) split by the classifiers is shown in Table 4.

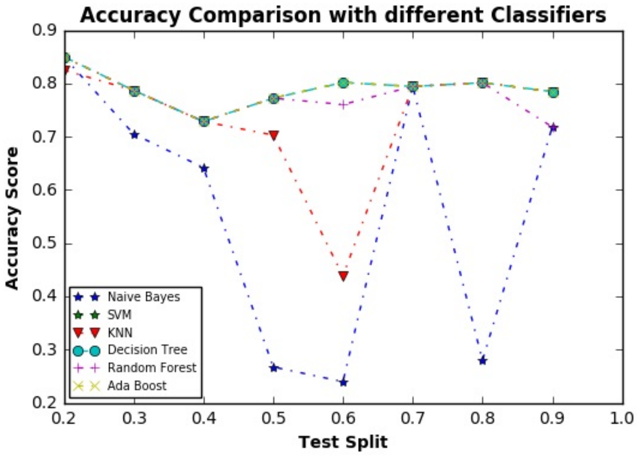


Figure 15: Accuracy of combined Algorithm of different classifiers

Table 4: 5 trials of the classifiers with a (80-20) split

Trial	NB	SVM	DT	KNN	RF	AB
1	0.725	0.8	0.8	0.8	0.8	0.475
2	0.625	0.775	0.775	0.775	0.775	0.775
3	0.3	0.725	0.725	0.725	0.725	0.725
4	**0.85**	**0.85**	**0.85**	**0.85**	**0.85**	**0.825**
5	0.75	0.85	0.85	0.85	0.85	0.85

Table 5: List of Classification Approaches

NB	Naive Bayes
SVM	Support Vector Machine
DT	Decision Tree
KNN	K-Nearest Neighbor
RF	Random Forest
AB	Ada Boost

4.3 Experimental Evaluation

Experiments were conducted on the algorithms for sarcasm detection with 188 Telugu conversation sentences as a testing set. The testing set consists of a 50:50 ratio of sarcastic and non-sarcastic conversation sentences *i.e.* 94 sarcastic sentences and 94 non-sarcastic sentences as ground truth. The experimental result in the form of confusion matrix over 188 testing sentences is given in Table 6. Further, precision, recall, and F-score are given in Table 7.

Table 6: Result of Proposed Algorithms in terms of Confusion Matrix

	T_p	F_p	T_n	F_n
Combined all three algorithms (188)	87	4	90	7
Only TNWS algorithm (188)	47	7	126	8
Only TIWS algorithm (188)	20	1	166	1
Only RiFoQ algorithm (188)	6	3	177	2

Table 7: Result of proposed algorithms in terms of Precision, Recall, F-score

Algorithms	Accuracy	Precision	Recall	F-score
Combined all three algorithms	90.5%	0.876	0.923	.899
Only TNWS algorithm	91%	0.741	0.6969	.718
Only TIWS algorithm	88.5%	0.84	0.851	.846
Only RiFoQ algorithm	91.5%	0.653	0.68	.666

5 Conclusion

In the area of sarcasm sentiment detection in low resource domain like Hindi, Telugu, Tamil, Arabic, etc., little work has been done. The reason behind is the scarcity of datasets for analysis and experiment. The collection of the dataset in this domain is the biggest challenging task. In this article, we built a dataset of Telugu conversation sentences manually from videos and annotated as sarcastic sentences. To identify sarcasm in collected dataset, we proposed a set of algorithms. There is no reported work on Telugu sarcasm detection so far. Therefore, these algorithms make an initiation in this direction. The proposed algorithms attain an accuracy of 94.14% with the limited amount of Telugu conversation datasets.

Acknowledgement: The authors would like to thank Bala Prakash, Vijay Chintala, and Madhusudan for providing annotation of our collected data set. All the annotators belongs to the states of Andhra Pradesh and Telangana.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] N. Desai and A. D. Dave, “Sarcasm detection in hindi sentences using support vector machine,” *International Journal*, vol. 4, no. 7, 2016.

[11] O. Tsur, D. Davidov, and A. Rappoport, “Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews,” in *Proceeding of International Conference on Weblogs and Social Media*, 2010, pp. 162–169.

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

[13] J. W. Pennebaker, M. E. Francis, and R. J. Booth, “Linguistic inquiry and word count: Liwc 2001,” *Mahway: Lawrence Erlbaum Associates*, vol. 71, no. 1, pp. 1–11.

[14] C. Strapparava, A. Valitutti *et al.*, “Wordnet affect: an affective extension of wordnet,” in *Proceedings of Language Resources and Evaluation Conference*, vol. 4, no. 1, pp. 1083–1086.

[15] D. Bamman and N. A. Smith, “Contextualized sarcasm detection on twitter,” in *Ninth International AAAI Conference on Web and Social Media*, 2015.

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

[17] A. Khattri, A. Joshi, P. Bhattacharyya, and M. J. Carman, “Your sentiment precedes you: Using an author’s historical tweets to predict sarcasm,” in *6th workshop on computation approaches to subjectivity, sentiment and social media analysis (WASSA) 2015*, 2015, p. 25.

[18] A. Joshi, V. Sharma, and P. Bhattacharyya, “Harnessing context incongruity for sarcasm detection,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 2, 2015, pp. 757–762.

[19] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, “Sentiment strength detection in short informal text,” *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010.

[20] J. Cohen, “A coefficient of agreement for nominal scales,” *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[21] 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.

[22] S. Reddy and S. Sharoff, “Cross language pos taggers (and other tools) for indian languages: An experiment with kannada using telugu resources,” in *Proceedings of IJCNLP workshop on Cross Lingual Information Access: Computational Linguistics and the Information Need of Multilingual Societies.*, Chiang Mai, Thailand, 2011.

[23] A. Bharati, R. Sangal, D. M. Sharma, and L. Bai, “Anncorra: Annotating corpora guidelines for pos and chunk annotation for indian languages,” Technical Report (TR-LTRC-31), LTRC, IIIT-Hyderabad, Tech. Rep., 2006.