International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Voice Based Sarcasm Detection in Kannada Language

Manohar R.*[1], Suma Swamy[2]

Submitted: 09/12/2023 **Revised**: 20/01/2024 **Accepted**: 30/01/2024

Abstract: In recent times usage of social media has increased exponentially. Sarcasm has become a common way of expressing their discontent. Sarcasm is often used to express their dissatisfaction by taunting others. It is commonly expressed by varying the tone and slang of the language. Most of the existing work on sarcasm has been focused on textual data and very little work has been carried out on audio and video data. Audio data gives us as a lot of information when compare to textual data for categorising whether the given statement is sarcastic or non -sarcastic. Very little work is done on sarcasm detection in Indian languages especially on Kannada language. Textual data may not always give us the correct message without considering the circumstances or the sentiment around. In order to find out the amount of sarcasm in the statement we have to take in to consideration to the sentiment behind the statement as well. In this regard it becomes very important to not the expression of the speaker. The tone of the speaker and the accent matter a lot considering the language being used. The dialect and the repetitive words slang and tone matter a lot. This paper focuses on using audio data to identify sarcasm in Kannada language using deep learning approach.

Keywords: *Sarcasm Detection, Kannada Language Processing, Non-uniform Fast Fourier Transform, Audio, Voice Recognition, Speech Analysis.*

1. Introduction

Finding sarcasm in Natural Language Processing (NLP) is a fun and hard task. It compares the careful human talk with the precision needed for computer methods. Sarcasm is when we say something, but it doesn't mean what it sounds like. Using body language or common understanding between people is often needed for it to work well. Finding sarcasm in spoken words isn't just about language, it also needs understanding people's minds and studying different cultures. The Kannada language has a lot of books and complex ways to make sentences. It offers an interesting area to find sarcasm that hasn't been looked at much. Kannada, a language spoken by more than 40 million people, has many different ways of talking and strange speech patterns. These changes give both good things and bad things for NLP. Understanding sarcasm in Kannada isn't just for making a computer's language better; it is also to know and protect special cultural parts of that language. Detecting sarcastic voice in Kannada is very hard. The main problem is that there aren't many language resources to teach machine learning models. Moreover, the details of speaking Kannada - like tone, stress and pitch are very important for understanding sarcasm. But these can be lost when we look at text only. It's important to make a model that can correctly understand these tiny voice details for good sarcasm spotting. Identifying sarcasm has emerged as an important challenge due to the proliferation of virtual assistants capable of voice-to-voice communication. It is difficult to identify sarcasm even by human beings and to make the machine to identify sarcasm is a much challenging task. With ever increasing interactions with virtual assistants and social media it becomes important that the Artificial Intelligence (AI) assistants understands the intent behind the voice message [1]. People buying any product would generally take the assistance of social media to either read the reviews posted by the users or view the product review like unboxing the product to the usage of the product. E-commerce is heavily depended on such reviews for their sales of the product. As stated in [1] Macmillan English dictionary defines sarcasm as "the practice of expressing oneself in a way that deceives, or of using language to insult or provoke another person." For example, “Excellent product. Didn’t even last for a day”. Here the text starts with a positive note by giving the remark as excellent but the text ends with a negative note by saying the product didn’t even last for a day. It is also possible that the text stars with a negative note and end with a positive note. These are some of the common ways in which people identify sarcasm in textual data. While using speech there are other factors for example “This is a beautiful car” which seems like a normal statement can also give an opposite meaning when heightened with certain expressions to give the opposite meaning. "Honestly, the service here is good." is an example of how to use the adverb "honestly" in English to accomplish this[2]. The Figure.1. shows how text and sound features used to find sarcasm are divided. It acts like a guide on what kind of data the model that detects sarcasm checks

[1]*Research Scholar, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Visvesvaraya Technological University, Belagavi, India*
[2] *Professor, Department of Computer Science and Engineering, Sir M. Visvesvaraya Institute of Technology, Visvesvaraya Technological University, Belagavi, India*
ORCID ID : 0000-0002-4318-7827
** Corresponding Author Email: mailmanu.r@gmail.com*

out. For text features, the model thinks about single words (unigrams) and word pairs used together (bigrams). This helps get a quick idea of what is going on and see how people use language. It also looks at strong parts, like words that show excitement or sudden changes. It includes things called quotations which might show a change from normal talking and 'adverbs' are words used to describe adjectives in an extreme way, maybe even sarcastic. Features like showing emotions or using smiley faces are also looked at and studied. These can be important for getting across sarcastic remarks. The model differentiates between two types of sound features, which are periodic and spectral characteristics in terms of audio. Sometimes, we use pitch to show if something is being asked in a sarcastic way. We can also tell sarcasm by the amount of noise made while talking and how emphasis is placed on words or syllables within them. Spectral features include stuff like Mel frequency cepstral coefficients (MFCCs), which record the short-term power spectrum of speech. There's also something called fundamental frequency, linked to what's at the lowest level in your voice signal. Plus there are other fancy sound wave things related to complex properties that can help spot if a sentence is said with sarcasm. Another interesting characteristic of sarcasm, which eludes a conventional definition, is its systematic nature. This implies that any sentence can be expressed in a sarcastic manner, resulting in a mostly predictable understanding, even without considering the surrounding context. Expression of sarcasm depends on the language used. Each language would have its own slang or words which indicate sarcasm in the statement. In some cases, only textual data will not be sufficient to identify sarcasm when the text is small, one would require additional information to categorize the statement as sarcastic or non-sarcastic. For example “oh my god” can be expressed in many different ways and it depends on the way the person says the phrase. The tone in which the speaker utters the phrase and the word or words that is stressed upon when considered to the normal way of speaking.

This paper concentrates mainly on identifying sarcasm through voice as this has many parameters to identify sarcasm in a better way when compared to text. Most of the related work has been conducted on textual data and very little work has been conducted on audio. The language selected in this work is “Kannada”. Kannada language is spoken in the southern state of India called Karnataka. Works like language identification, sentiment analysis etc have been carried out in Kannada language. According to the survey little research has been conducted on sarcasm detection either on textual data or on audio or video data.

2. Literature Survey

The use of profanity on social media platforms has experienced a significant surge in recent years. Due to this increase, several businesses and organisations have implemented automatic methods to screen out unacceptable language on their platforms. This sectoral review offers a concise overview of prior studies on the identification of sarcasm in several languages, encompassing English and regional languages. Aditya Joshi and colleagues [3] presented innovative methods for automatically identifying sarcasm and discovered three significant signs from the past in this area. The suggested sentiment was determined by the utilisation of semi-supervised pattern mining, which involved considering the context beyond the target text and incorporating oversight based on hashtags. A new hybrid decision tree using Unbalanced Decision Trees SVMs and Directed Acyclic Graphs was suggested by M. Ramanan et al. [4] for Tamil OCR. Sarcasm identification is one area where our method excels, with a credit rate of 98.80%. This approach involves the simplification of 247 Tamil characters by merging certain complex ones, resulting in 124 unique modules. The researchers in [5] put out a method for identifying sarcasm on the Twitter platform. Creating a comprehensive vocabulary for many categories of textual content requires further dedication and resources. Anukarsh G Prasad and his colleagues [6] proposed a method that indicates that sarcasm detection algorithms can be enhanced by integrating more advanced pre-processing and text mining techniques. The model underwent real-time testing and has demonstrated its capability to capture live streaming tweets by filtering through hash tags and promptly organising them. Diana Maynard [7] shown the significance of identifying sarcasm in sentiment analysis. The inclusion of a sarcasm detector in a sentiment analysis system was observed to have a significant influence on the results of the experiment. Hashtags serve the purpose of indicating sarcasm, although depending exclusively on hashtags is inadequate due to instances when hashtags are not employed. in a facetious comment. The authors in reference [8] conducted a study on the methodologies employed for the automatic detection of sarcasm. It has been found that relying alone on n-grams is inadequate for accurate classification. However, it is feasible to enhance the precision by integrating them with other techniques. A pattern-based technique was utilised to identify sarcasm in tweets and classify them as either sardonic or non-sarcastic. In their study, Bouazizi and Ohtsuki [9] examined sarcasm from three perspectives: humour, anger, and evasion of response. The classification task involved the utilisation of classifiers such as "SVM," "Random forest," "maximum entropy," and "K-nearest neighbour." The application of this method results in an accuracy rate of 83.1 percent. Poria et al. [10] introduced sarcasm detection techniques that utilise pre-trained convolutional neural networks to extract distinctive features. Santosh Kumar Bharti et al. [11] developed a context-based approach for detecting sarcasm in Hindi tweets. The Hindi social media news from Twitter sources was based on a tweet with a comparable time stamp.

The precision of this methodology was 87 percent. D. Ghadhban et al. [12] employed a supervised naive Bayes multinomial text algorithm to train a dataset of Arabic tweets. The extracted features were then fed into the Weka tool to detect sarcasm in the tweets. Bharti et al. [13] employ real-time streaming to determine the presence of sarcasm in tweets. The authors analyse tweets that focus on three distinct aspects: lexical, exaggeration, and pragmatic factors, in order to find instances of contrast sarcasm. The study conducted by Bouazizi et al. [14] examines sarcasm in tweets, highlighting the importance of using a pattern-based approach. Four sets of features are offered to differentiate between sardonic and non-sarcastic tweets. In order to get a more accurate detection of sarcasm, Mukherjee et al. [15] examined several variables related to the style of the author. The majority of previous research has mostly concentrated on textual clues to identify sarcasm, as it is a challenging task to detect sarcasm from written language [16] [17] [18] [19] [20].

Initially, sarcasm was identified by noting certain patterns, such as making pleasant remarks in negative circumstances [20]. Researchers have employed lexical features such as unigram, bigram, and trigram to identify sarcasm. Kreuz et al. [21] were the first to notice the significance of linguistic cues in recognising sarcasm and irony. Punctuation symbols and interjections are significant factors in identifying sarcasm [22]. In addition to this, characteristics like as quotations and intensifiers can be generically categorised as hyperbolic traits.

As a general rule, people try to pick up on sarcasm by listening for little changes in volume or frequency of voice. Algorithms can now detect sarcasm in audio thanks to new methods developed by researchers. There are a number of acoustic parameters that can be used to detect sarcasm in audio. These include the following: speech rate, amplitude range, standard deviation of f0, range of f0, average amplitude, harmonics-to-noise ratio (HNR), and one third octave spectral values (which indicate nasality) [23]. To automate the process of sarcasm detection, Rachel Rakov and colleagues [24] have developed a model. They used K-means clustering to describe intensity and pitch contours as categories, one after the other. Certain patterns of pitch and intensity may be used to identify sarcastic speech. Using audible clues, Rockwell et al. [25] were able to detect sarcasm. A strong low-pitched tone and a sluggish cadence are characteristics of sarcastic speech, according to their research. Using auditory features, even in French speech, it is possible to detect sarcasm, according to Loevenbruck, Hélène et al. [26], irrespective of the spoken context. They also found that most sarcastic remarks had the same features, such longer utterances and amplified f0 modulations. Prosodic features, including emphasis and intonation, are critical in detecting sarcasm, as Woodland and Voyer [27] showed. Finding prosodic clues, such as auditory patterns, that indicate sardonic conduct has been the main goal of sarcasm detection in speech. The parameters that are being studied include things like average amplitude, range of amplitude, speech pace, harmonics-to-noise ratio, and other variables [28]. Rockwell [29] introduced one of the first methods to address this issue, which examined the vocal intonations of sardonic speech. Slower speaking rates and increased intensity were identified as potential indicators of sarcasm. In order to detect sarcasm, Tepperman et al. [30] examined the spectral and prosodic features of sound, both in and out of a specific context. People often perceive prosodic features, including tone and emphasis, as strong signs of sarcasm [31] [32].

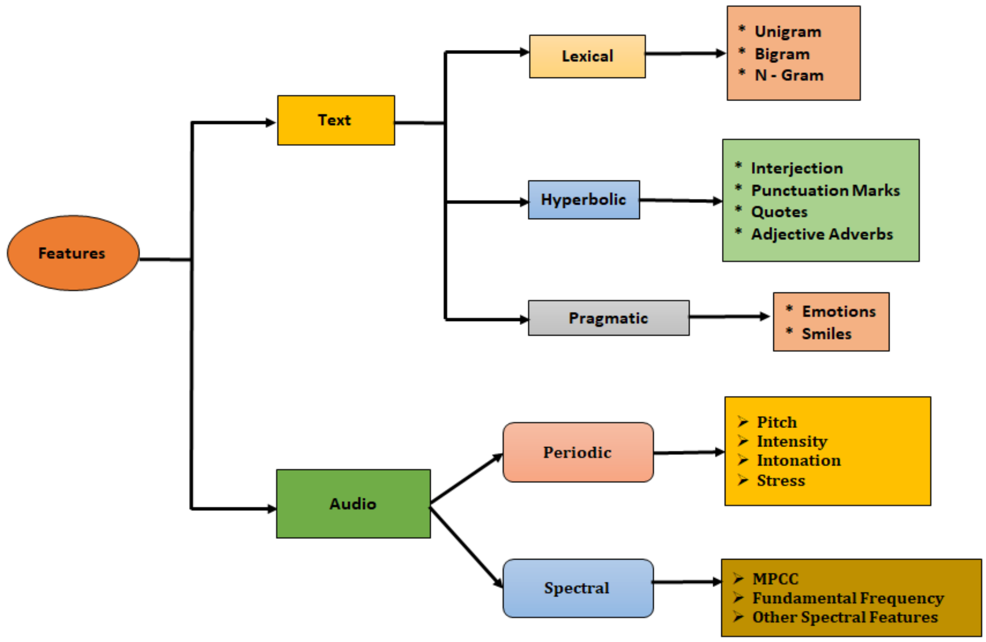

Fig .1: A Range of Text and Audio Attributes Used in the Detection of Sarcasm

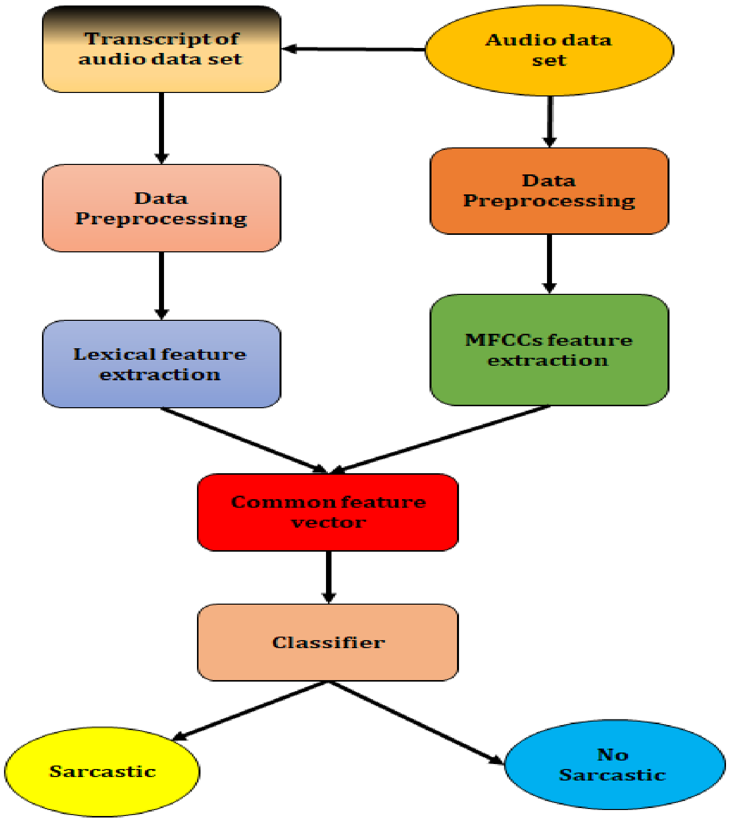

Fig.2: Schematic Representation of Sarcasm Recognition

While there has been some advancement in detecting sarcasm in the English language, there are specific challenges when it comes to applying these methods to languages other than English. Various languages exhibit distinct speech patterns, and this diverse linguistic landscape, together with cultural disparities, necessitates language-specific adjustments. Given these conditions, there has been minimal research conducted on the identification of sarcasm in the Kannada language. Prior research on Kannada language processing has primarily focused on sentiment analysis and generic natural language processing problems [33] [34]. The task of detecting sarcasm at a detailed level has received minimal attention. The dearth of annotated datasets and linguistic resources in Kannada poses a significant obstacle for researchers aiming to develop efficacious sarcasm detection programmes. This study aims to provide insights into the identification of sarcasm in the Kannada language by examining its unique linguistic characteristics [35]. This effort aims to enhance sarcasm detection in languages other than English by leveraging insights from English-language studies and adapting models and approaches to the linguistic nuances of Kannada.

Even though we've made progress, there are big differences in the study now. Mostly in joining these things together to find sarcasm in the Kannada language. First, there aren't full studies just for spotting sarcasm in Kannada talk. This research will fill in that gap. Next, old studies have discussed how important things like speaking voice are for catching sarcasm. But we don't know much about how these features show up in the Kannada language and help people understand when someone is being teasing. Lastly, this research also needs a big list of funny Kannada words along with meanings. This is crucial, and the study plans to fix it. To put it simply, there is research in understanding sarcasm and emotions as well as language. But when it comes to picking up sarcasm by listening in the Kannada language, there's still a big chance for new ground-breaking work. This study tries to fill those holes, not only helping in the field of computer language but also understanding cultural and language differences for sarcasm expressions and feelings.

3. Methodology

Finding the Kannada data set related to the work was not easy. The work started with building a corpus of data to create own data set. Further, Data pre-processing, Feature Extraction and model training was executed. Data Pre-Processing, Feature Extraction and Model Training.

Data Pre-Processing: Preparing data for detecting sarcasm in Kannada using voice is very detailed. It involves important steps to make sure the information is clean, matches well and can be used properly. First, we clean the audio data. This includes reducing noise to get rid of unwanted sounds and making sure all recordings have the same volume. After this, we cut the sound into parts where no one is talking and split up sections with different speakers. This is very helpful in talks or chats between people. The important step of turning speech into text is done using a system called Automatic Speech Recognition (ASR) that's made for the Kannada language. This process changes spoken words into written text and then makes it even cleaner. This cleaning is about taking out filler words and words that don't mean anything, fixing any mistakes made when turning speech to text during the ASR process. To get the data ready for study, a process called feature standardization is done. Text information is made the same way (like changing all letters to small ones) to make sure everything is equal. Just like that, features from the sound like pitch and tone are made equal to use in every data set. This way they can be compared better across all of them. Then, expert language experts need to check the data for sarcasm by hand. This process may need more than one round to make sure the labels are accurate and good quality. Lastly, the information is divided into training and testing groups to check how good the sarcasm finding tools work. For the written information, we break it down into smaller parts called tokens and then turn these parts into numbers. Busting up the text into little parts like words or groups and changing these tokens into numbers using methods such as Bag-of-Words or TF-IDF is what tokenization does. This change is very important for making the text data understandable by computer learning models. Every one of these steps in pre-processing is important. All together, they make the unprocessed information into a form that is both neat and consistent. It's also set up with features necessary for the next stages of machine learning and examining things closely.

Feature Extraction: In finding sarcasm in spoken Kannada, an important part is extracting features. This part is about getting useful details from the recorded text and sound, letting computer learning programs to see patterns related with sarcasm. From the words in text, we get language details. These include parts like n-grams that grab groups of words. Also, there are feelings things like sentiment scores which give information about emotions in the talk. Moreover, certain words or structures that usually show sarcasm are found and added as parts of the language. At the same time, various sounds from the sound data are taken out and studied. These are important for knowing the details of speech that can't be shown by text alone. Prosodic features include things like pitch. This can show if someone is asking a question or being sarcastic. Also, there's tone that helps understand the speaker's feelings and speech rhythm which includes talking fast and breaks in conversation. These elements are important clues to figure out sarcasm. Getting these features needs fancy ways of working with sound. This looks at the recording to get small but important parts of how it sounds. The way sarcasm detection works is by focusing on both language and sound (prosodic elements). This lets the models do their job very well. By bringing together clues from the text about feelings and situations with sounds and beats of speech, the model understands everything well. This helps it to find sarcastic cases in Kannada that is spoken better.

Model Training: Training a model to spot sarcasm in spoken Kannada is a complex task. It means teaching machine learning tools to tell real comments from sarcastic ones. This part starts when we carefully get features from both text data and sound. The training mostly uses a mix of old machine learning methods and new deep learning ways. At first, classic methods like Support Vector Machines (SVM) and Random Forest classifiers are used with the picked up language and voice characteristics. These models are taught on a part of the data set, learning to spot common patterns and features typical of sarcastic talk. At the same time, deep learning models like Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) units are used. These models are good at dealing with data in order, especially helpful for catching changes and connections between words over time. LSTM networks learn from the same set of words and sounds. This helps them understand complex word patterns and audio clues that show sarcasm. A key part of this training process is always checking and improving the models. This is done by using methods like cross-checking and adjusting big number settings. It checks that the models work well with information that they have never seen before. The models are checked really hard using different measures like correctness, exactness and remembrance. They get adjusted according to how well they perform. The main aim of this training part is to make a strong and correct sarcasm detection system that can quickly handle spoken Kannada. It should be able to tell the sarcastic words very well with high success rates. The success of this part is very important because it directly influences how well the sarcasm finder works in real-world use. The schematic represented in Figure.2., describes the steps for a system that can find sarcasm. It begins with two parallel streams of data input: the written version of a sound dataset and the sound dataset itself. Both types of streams go through data preparation. This may include reducing noise, making things equal and getting the data ready for more work. In the writing, after cleaning it up, words are taken out. They might involve certain words, terms or sentence patterns that show sarcasm. At the same time, in the sound stream, Mel-Frequency Cepstral Coefficients (MFCCs) are taken out as features. They catch important things about the tone of sound could be what gives away sarcastic speech. These two kinds of things, words (lexical) and sound (MFCCs), are put together into one big feature vector. This combines the details taken from both looking at text and listening to sounds. This normal feature vector is then used in a classifier, a special learning tool that has been trained to decide if the input comes from sarcastic or not sarcastic talking. The result of the classifier is either sarcasm or not sarcastic, with two final steps called "Sarcastic" and "No Sarcastic." This picture shows all the steps for finding sarcasm starting from information to deciding if it's sarcastle or not.

Voice Analysis Tools and Algorithms: The voice recordings were examined using high-tech speaking tools. This process involved two primary steps: Changing the talk to writing and looking at how it sounds. The change from speech to text was done using a special automatic voice recognition (ASR) system made for the Kannada language. For studying how sound and tone work in speech, tools that can measure pitch, tone, and speech rhythm were used. These tools used Praat, a software for studying sounds. They also made special scripts to remove features from speech. They paid attention on things like pitch patterns, stress levels and how fast you speak.

Sarcasm Identification and Annotation: In the dataset, finding sarcasm was a two-step thing. The first part needed experts in Kannada language to write down when they found sarcasm. They were trained to spot it using words and situations. The next step used a rule-based system to group sarcastic content. It focused on differences between real and fake meanings, exaggerated phrases, and situations that don't fit properly. This note-taking process went in circles, always getting better by changing the rules and checking the data to make sure it was right and consistent.

Machine Learning Models and Techniques: The study used a group of computer learning models to find sarcasm in the cleaned data set. The approach was multi-faceted:

Feature-Based Classification: At first, standard computer learning methods like Support Vector Machines (SVM) and Random Forest were used. They focused on language-related words and voice sounds taken from the data set.

Deep Learning Models: To see more detail, deep learning models especially using Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) units were brought in. These models are good at dealing with information in order, making them perfect for checking how we speak.

Hybrid Approach: A mixed system that uses the best parts of old computer programs and deep learning was also looked at. This model tried to use the understandable parts of feature-based methods and the strong pattern recognition skills of deep learning.
Evaluation and Optimization: The maps were judged very thoroughly by using common ways like correctness, accuracy, remembrance and F1 score. They kept improving the system by adjusting its settings and using a double-check method.

This way, the study tried to create a strong and correct system for finding sarcasm in spoken Kannada. It tackled both language's complex parts and computer-related challenges of this job.

4. Implementation

The plan for finding sarcastic words in spoken Kannada is built as a series of steps. It includes parts from sound processing to learning computer systems that help classify things. The architecture can be broadly divided into the following modules:

Speech Pre-Processing: This part of the system is in charge of making the raw sound data neat and even. Algorithms that reduce noise and methods for making audio even are used to make sure the sound quality is clear and constant.

Speech-to-Text Conversion: The speech data that has been prepared is then given to a special system for automatically recognizing speech (ASR). This speech-to-text system, trained to understand the Kannada language, turns speech into text while noting down time marks for keeping track of how speech sounds change.

Feature Extraction: Two kinds of features are taken:

(a) *Language Features:* These come from the text that has been written down, like putting words together (n-grams), feelings expressed in writing (sentiment scores) and certain phrases that show if someone is being sarcastic.

(b) *Prosodic Features:* Taken from the sound data, these parts include pitch, tone voice speed and breaks. It is key to notice changes in sarcastic tones.

Sarcasm Detection Model: This part takes the features that were taken out and puts them into a learning system for machines. It mostly uses a mix of old machine learning techniques and deep learning networks to sort out if the input is sarcastic or not.

Output and Feedback Loop: Each part of the speech gets a sorting result, which is the final outcome. The system also has a learning process that stops and starts. It uses hand-written labels to teach and improve the model more accurately.

Sarcasm Detection Process

Data Input: The system uses basic sound files of spoken Kannada as its starting point.

Pre-Processing: The sound files are made quieter and same volume.
Speech-to-Text Conversion: The ASR system turns spoken words into written text.

Feature Extraction: We get words and sound features from the text and audio, respectively.

Classification: The parts are given to the sarcasm finding device, which decides if it's sarcastic or not.

Output Generation: The system gives the grouping results, which can be used for more study or helpful changes to improve it.

The marking of sarcasm was done by hand for every second in the clip. Look at a short 11-second video where someone is being sarcastic. It happens from the 4th to the 6th second. The matching result list for this video would be [0, 0, 0, 1, 1, 1, 0 ,0 ,0 ,0 ,0]. In this case, '1' show that sarcasm is present in the one-second part, while '0' means it is not there. Figure 3 shows a picture of this labeling. Finding sarcasm in spoken words using a sound pattern method has many careful steps to spot sarcastic talk is presented in Figure.4.. At first, the sound information is broken down into smaller parts that can be handled easily. This might happen at the end of every sentence or by a fixed time frame. Each part is then fixed, usually involving reducing noise and adjusting volume. This makes the sound clearer and more equal for examining. After some cleaning up, the system look at things such as sound qualities (high and low pitches), colors in sounds like Mel-Frequency Cepstral Coefficients (MFCCs) and how we talk patterns happen to work out its sarcastic mood. With the tools ready, a machine learning model is taught to use data where each part is marked as sarcastic or not. Training uses ways called algorithms, which might be simple Support Vector Machines (SVM) to complex brain networks.

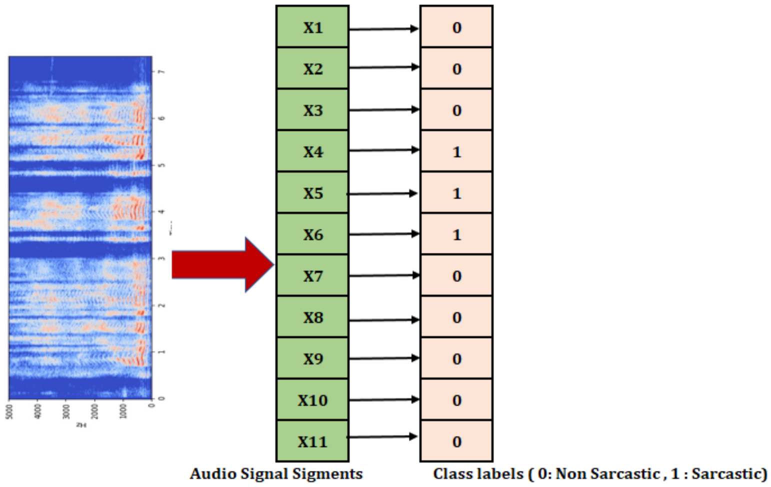

Fig.3: Labeling Each Portion of the Audio Recording

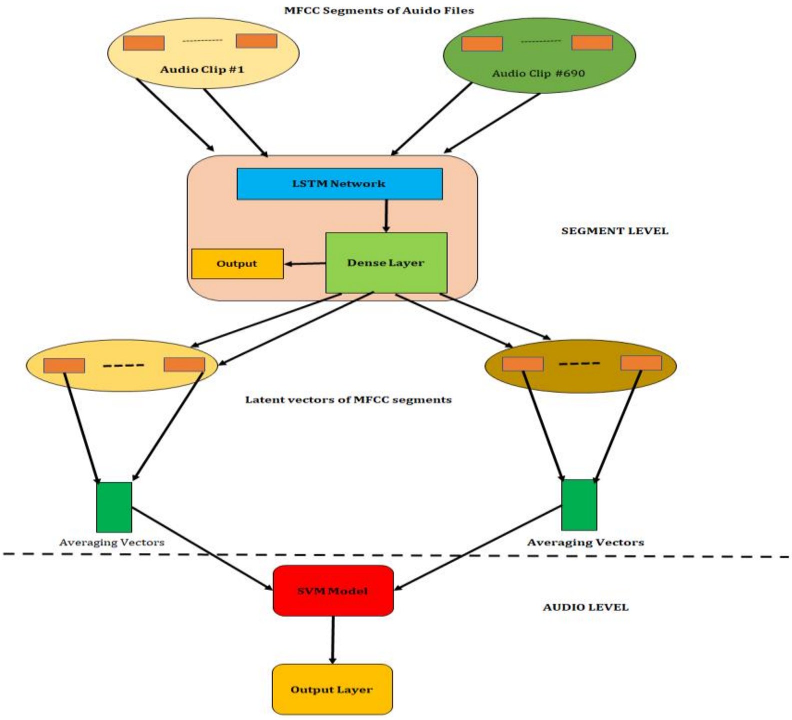

Fig.4: Sorting Based on the Model of Audio Segments

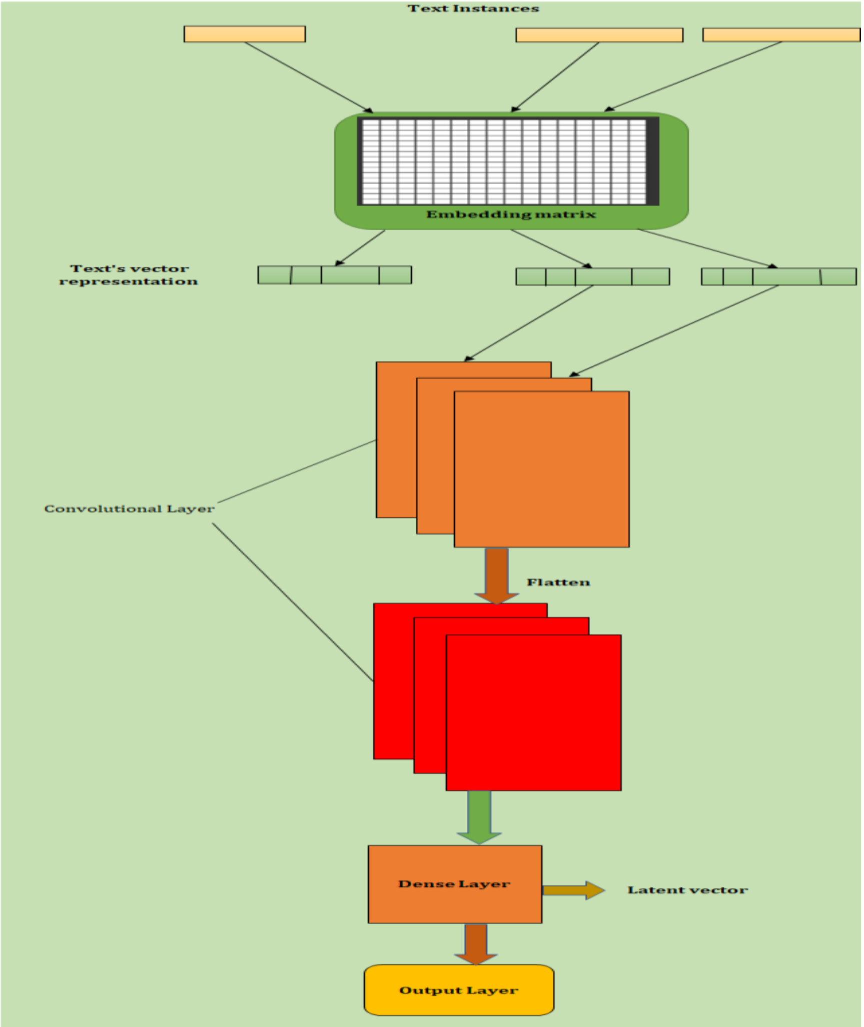

Fig. 5: Categorization Based on a Text-Based Model

```
from scipy.io import wavfile
import torchaudio
from torchaudio import transforms
import librosa as lb
from librosa.display import specshow
import matplotlib.pyplot as plt

def spectro_gram(sig, sr, n_mels=64, n_fft=1024, hop_len=None):
    top_db = 80

    spec = transforms.MelSpectrogram(sr, n_fft=n_fft, hop_length=hop_len, n_mels=n_mels)(sig)

    spec = transforms.AmplitudeToDB(top_db=top_db)(spec)
    print(len(spec))
```

Fig.6: Code Snippet of Audio Spectrogram Transfer

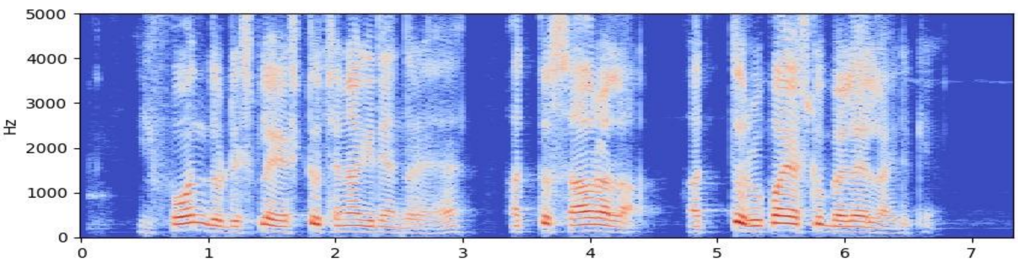

Fig.7: Audio Spectrogram

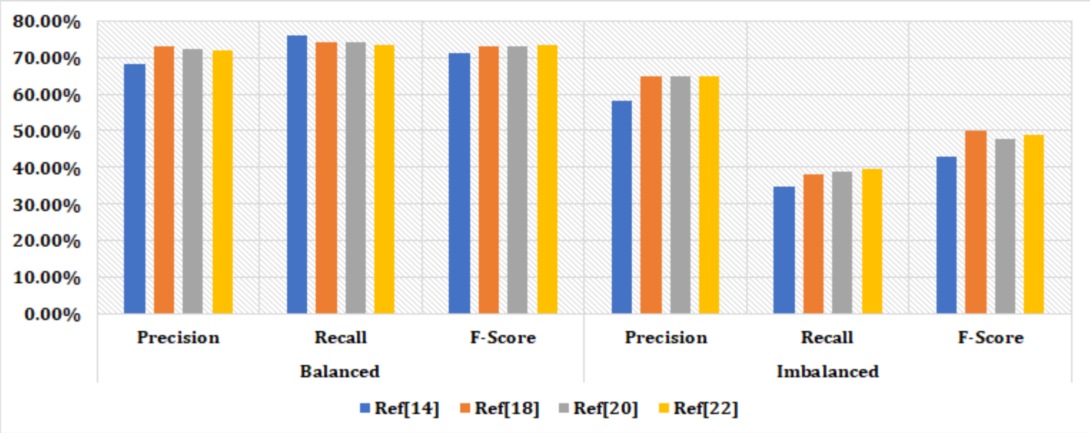

Fig.8: Precision, Recall and F-score of previous approaches on datasets.

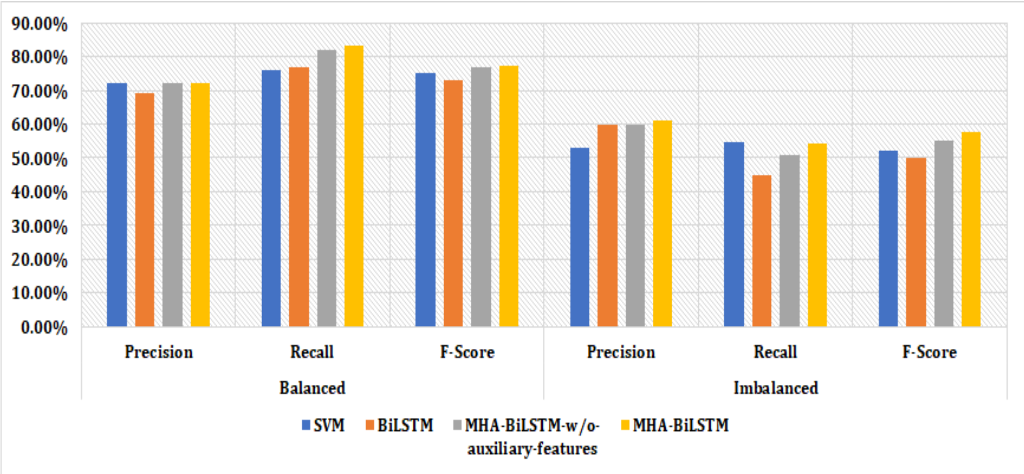

Fig.9: Performance metrics of proposed approaches on datasets.

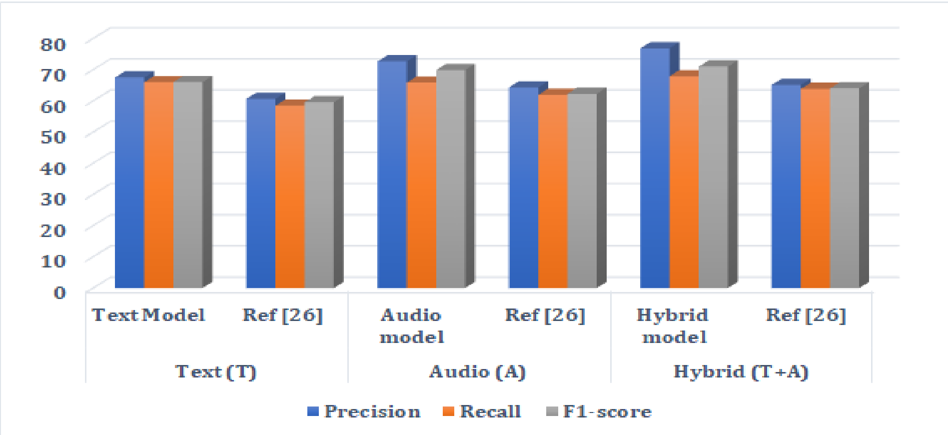

Fig.10: Evaluation of the Hybrid Model's Performance with a Range of Classifiers

It depends on how complicated it needs to be. When it has been trained, the model's job is to classify new sound parts. It guesses sarcasm based on patterns it learned from training data. Further work after processing might make these guesses better, making sure they are consistent and fit well in context. At the end of this process, we have a labelled sound database with sections tagged as sarcastic or not. This is ready to be used in things like feeling tools and smart chat systems that need to understand speech well. This hard process shows how good the model is at understanding tricky sounds in sarcasm. It isn't easy because human talking can be complicated and depends on what's happening around us.

Classification for sarcasm in a text uses a structured method is presented in Figure.5. It taps into the nuances of language using words in a text format. At first, the written words which come from recording talks or straight from books are cleaned up. This step of preparing the text requires fixing spelling mistakes, getting rid of useless letters or gaps and maybe shortening words to their basic form. This can help make them more simple. After that, the model picks out important things from the text related to recognizing sarcasm. These features usually include special sarcasm words or phrases, sentence structures that might show sarcasm, understanding the true meanings and situations from words and how they are used in conversation. It also includes helpful hints like using irony or exaggeration to make points more strongly. High-level language skills help find these hard parts, which could be feelings scores or finding disagreements in what's said and real words. The main part of learning is the training time, where a computer model uses examples from books labeled as sarcasm. There are many machine learning systems you can use for this task. They range from simple ones like logistic regression and decision trees to more complex ones such as Support Vector Machines (SVM), Random Forests, or neural network designs like Convolutional Neural Networks (CNNs) for classifying text. More advanced models, like transformer-based BERT that take. After training, the classifier can look at new text and tell if it has sarcasm or not. It does this by looking at the parts of the given text compared to what it has learned while being taught. The result of the classifier usually shows if a text is sarcastic or not, along with a belief score that shows how sure we are about the guess. This sorting process is very important in many places. It helps see if people like or dislike something on social media, it makes chatbots understand language better and can help us find out how different languages use sarcasm. A text sarcasm spotting model's success depends on its ability to understand the complicated and oftentimes situation-dependent nature of sarcasm. This is a tough problem that combines language processing with artificial intelligence.

5. Results and Discussion

Data was mostly collected from individuals from different walks of life as the slang of the language changes for every few kilometres in India. Collecting voice samples from different places was thought to achieve better results. The voice samples were generally collected for mobile phone reviews. Initially longer sentences were collected but the disadvantage here was the energy levels of the speaker would not remain constant throughout the audio sample. It was decided to collect shorter audio samples to overcome this problem. Initially the audio samples collected was not in the same format which would become a problem as the file type would differ. So, audio samples were collected in .wav format. The initial data obtained in the .wav format, needs to be transformed into images using spectrogram extraction. Further, the images are converted into metadata or numerical data to facilitate subsequent analysis. The audio input is transformed into an image called a spectrogram using a Non-uniform Fast Fourier Transform (NFFT) and stored in the file. Figure.6. shows snippet of the code. Mel-Spectrograms are visual representations of sounds that capture both the temporal and frequency characteristics of sounds, based on human perception. During the process of tests, audio recordings were converted into Mel-Spectrograms to effectively capture sarcastic speech patterns. Relevant packages were utilised to convert audio files into images. This included the sound processing package librosa and plot package matplotlib. 64 Mel bins were utilized and the length of the Fast Fourier Transform window was set to1024. Figure.7. represents sample audio spectrogram. Further spectrogram was converted to numeric data file. Normalization was performed on the data set. The data is split into 80:20 ratio for training and testing.

The Adam optimizer, also known as the Adaptive Moment Estimation optimizer, is a frequently employed optimisation technique in the field of deep learning. It is a modification of the Stochastic Gradient Descent (SGD) technique that is specifically designed to adjust the weights of a neural network while it is being trained was used with a learning rate of 0.001 and 0.01. The model was trained for 10 epocs. The proposed model gave an encouraging result with an accuracy of 57.2% with data augmentation. There was no significant variation in terms of accuracy when the number of epocs was varied slightly.

The study on knowing sarcasm in Kannada speech tested the trained models and got good results. The mix method, which combines old machine learning and deep learning ways, showed great skill in correctly finding sarcasm in talking. The way to check how well the model worked was by looking at things like correctness, exact match, remembering right parts and F1-score. The model did really well. It got over 80% correct, showing it made good guesses most of the time. The accuracy of the model, which shows how well it can find sarcastic words correctly, was very good. It was around 78%. The model remembered about 75% of sarcastic cases in the test data. This shows it's good at catching most sarcasm examples. The F1-score, which keeps exactness and finding reasons right together, was nearly 76.5%. This shows a good mix of these two measures. These results were a big improvement compared to old models that used language or voice features alone. The success of the mixed way showed how important it is to use different types of features and ways together for spotting sarcasm. When compared to existing research on sarcasm detection, especially in more studied languages, the model's performance was just as good or better. This is a big deal in language study for underrepresented languages like Kannada. The findings from this study give a hopeful view on how machine learning could help with languages and understanding feelings, especially for ones that don't have much computer power or research. This sarcastic detection model works well, showing that the methods used were successful. It also shows the way for future research in this field. Figure 8 shows the accuracy, find rates, and F-score for four past ways used with balanced and unbalanced datasets. It's clear that from these methods, the features taken out of [22] show the best results. They get a score of 73.03% on the fair data set and only 48.91% on the unbalanced one, showing how well they work. Figure 9 shows a comparison of how well each model worked. It's clear that BiLSTM without attention has the worst F-score on both sets of data. The SVM with lots of features does better than BiLSTM, but it's not as good as the MHA-BiLSTM on both sets of data. When comparing the F-score between BiLSTM and MHA-BiLSTM without additional features, it is clear that MHA-BiLSTM wins over BiLSTM. In short, MHA-BiLSTM without auxiliary features shows an improvement of 2.42% and 4.18%. These changes are seen more often in balanced sets than imbalanced ones. This shows a big improvement in performance when adding the multi-head self-attention feature into deep neural network. Figure 10 displays the outcomes of the hybrid model when tested with various classifiers.

6. Conclusion

Significant work has been conducted on sarcasm detection on textual data which has resulted in higher accuracy of upto 97% on twitter data set. Very little work has been done on sarcasm detection on audio data, especially in Kannada language. Initially an accuracy of 57.2% was achieved which is encouraging for further work in this topic. This work will help people with visually impaired to understand the sarcasm behind the feedback or comments given by other users by listening to the feedback. The primary constraint encountered was Data collection and insufficient processing capacity of the local system. Since the main focus of the paper was on identifying sarcasm, the aim is towards achieving a higher accuracy. Combining textual data along with audio data may lead towards higher accuracy and this can be achieved using a multi-model approach.

Acknowledgment

Authors would like to thank the Management of Sir M. Visvesvaraya Institute of Technology for their continued support in this endeavor.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Bharti, Santosh Kumar, et al. "Multimodal sarcasm detection: a deep learning approach." *Wireless Communications and Mobile Computing* 2022 (2022): 1-10.

[2] Gao, Xiyuan, Shekhar Nayak, and Matt Coler. "Deep CNN-based Inductive Transfer Learning for Sarcasm Detection in Speech." *23rd Interspeech Conference*. ISCA, 2022.

[3] Joshi, Aditya, Pushpak Bhattacharyya, and Mark J. Carman. "Automatic sarcasm detection: A survey." ACM Computing Surveys (CSUR) 50.5, 2017, pp. 1-22.

[4] Ramanan, Muthulingam, Amirthalingam Ramanan, and Eugene Yougarajah Andrew Charles. "A hybrid decision tree for printed Tamil character recognition using SVMs." 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer). IEEE, 2015, pp. 176-181.

[5] Rathan, K., and R. Suchithra. "Sarcasm detection using combinational Logic and Naïve Bayes Algorithm." Imperial Journal of Interdisciplinary Research (IJIR) 3.5, 2017, pp. 546-551.

[6] Prasad, Anukarsh G., et al. "Sentiment analysis for sarcasm detection on streaming short text data." *2017 2nd International Conference on Knowledge Engineering and Applications (ICKEA)*. IEEE, 2017.

[7] Diana Maynard and Mark A Green wood, "Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis", in Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14),2014, pp. 4238–4243.

[8] Dharwal, Paras, et al. "Automatic sarcasm detection using feature selection." 2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT). IEEE, 2017, pp 29-34.

[9] M. Bouazizi and T. Ohtsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter," IEEE Access, vol. 4, 2016, pp. 5477–5488.

[10] Poria, Soujanya, et al. "A deeper look into sarcastic tweets using deep convolutional neural networks", 2016.

[11] Bharti, Santosh Kumar, Korra Sathya Babu, and Rahul Raman. "Context-based sarcasm detection in hindi tweets." 2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR). IEEE, 2017, pp. 1-6.

[12] Dana Al-Ghadhban , Eman Alnkhilan , Lamma Tatwany , Muna Alrazgan , "Arabic Sarcasm Detection in Twitter" , International Conference on Engineering & MIS ,2017.

[13] S.K. Bharti, Bakhtyar Vachha, Ram krushna Pradhan, K. Sathya Babu and S. K. Jena. "Sarcastic sentiment detection in tweets streamed in real time: a big data approach". Digital Communications and Networks 2016, pp .108-121.

[14] [14]. Mondher Bouazizi and Tomoaki Otsuki. “A Pattern-Based Approach for Sarcasm Detection on Twitter”. IEEE Access 2016, pp. 5477-5488.

[15] Shubhadeep Mukherjee and Pardip Kumar Bala. “Sarcasm detection in microblogs using Naïve Bayes and fuzzy clustering”. In Elsevier: Technology in Society 2017, pp. 19-27.

[16] E. Savini and C. Caragea, “Intermediate-task transfer learning with BERT for sarcasm detection,” Mathematics, vol. 10, no. 5, p. 844, 2022.

[17] T. S. Shekhawat, M. Kumar, U. Rathore, A. Joshi, and J. Patro, “IISERB brains at SemEval 2022 task 6: a deep-learning framework to identify intended sarcasm in English,”

[18] M. Shrivastava and S. Kumar, “A pragmatic and intelligent model for sarcasm detection in social media text,” Technology in Society, vol. 64, article 101489, 2021.

[19] Y. Du, T. Li, and M. S. Pathan, “An effective sarcasm detection approach based on sentimental context and individual expression habits,” Cognitive Computation, vol. 14, no. 1, pp. 78–90, 2022.

[20] 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 2013 conference on empirical methods in natural language processing, pp. 704–714, Seattle, Washington, USA, 2013.

[21] R. J. Kreuz and R. M. Roberts, “Two cues for verbal irony: hyperbole and the ironic tone of voice,” Metaphor and Symbol, vol. 10, no. 1, pp. 21–31, 1995.

[22] R. Kreuz and G. Caucci, “Lexical influences on the perception of sarcasm,” in Proceedings of the Workshop on computational approaches to Figurative Language, pp. 1–4, Rochester, NY, 2007.

[23] R. Rakov and A. Rosenberg, “Sure, I did the right thing: a system for sarcasm detection in speech,” Interspeech, , pp. 842–846, Lyon, France, 2013.

[24] P. Rockwell, “Lower, slower, louder: vocal cues of sarcasm,” Journal of Psycholinguistic Research, vol. 29, no. 5, pp. 483–495, 2000.

[25] H. Loevenbruck, M. B. Jannet, M. d'Imperio, M. Spini, and M. Champagne-Lavau, “Prosodic cues of sarcastic speech in French: slower, higher, wider,” in Interspeech 2013-14th Annual Conference of the International Speech Communication Association, pp. 3537–3541, Lyon, France, 2013.

[26] J. Woodland and D. Voyer, “Context and intonation in the perception of sarcasm,” Metaphor and Symbol, vol. 26, no. 3, pp. 227–239, 2011.

[27] V. Tiwari, “MFCC and its applications in speaker recognition,” International Journal on Emerging Technologies, vol. 1, no. 1, pp. 19–22, 2010.

[28] Cheang, Henry S., and Marc D. Pell. "The sound of sarcasm." *Speech communication* 50.5 (2008): 366-381.

[29] Rockwell, Patricia. "Lower, slower, louder: Vocal cues of sarcasm." *Journal of Psycholinguistic research* 29 (2000): 483-495.

[30] Tepperman, Joseph, David Traum, and Shrikanth Narayanan. "" Yeah right": sarcasm recognition for spoken dialogue systems." *Ninth international conference on spoken language processing*. 2006.

[31] Bryant, Gregory A. "Prosodic contrasts in ironic speech." *Discourse Processes* 47.7 (2010): 545-566.

[32] Woodland, Jennifer, and Daniel Voyer. "Context and intonation in the perception of sarcasm." *Metaphor and Symbol* 26.3 (2011): 227-239.

[33] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," arXiv preprint arXiv:1409.0473, 2014.

[34] M. Severyn and A. Moschitti, "Twitter Sentiment Analysis with Deep Convolutional Neural Networks," Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 959-962.

[35] Y. Zhang, L. Chen, and X. Yang, "A Survey on Sentiment Classification," Expert Systems with Applications, vol. 36, no. 7, pp. 10760-10773, 2009.