VISVESVARAYATECHNOLOGICALUNIVERSITY

BELAGAVI-590018

BLDE Association's

VACHANA PITAMAHA Dr. P. G. HALAKATTI COLLEGE OF ENGINEERING & TECHNOLOGY

VIJAYAPUR-586103

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

PROJECT REPORT
ON

"SARCASM DETECTION AND CLASSIFICATION ON REGIONAL LANGUAGE USING MACHINE LEARNING TECHNIQUES"

UNDER THE GUIDANCE OF

Prof. SANTOSH CHINCHALI

SUBMITTED BY

Ms. RUCHITA ATHARAGA	[2BL21CS186]
Mr. SACHINKUMAR MADARAKHANDI	[2BL21CS126]
Ms. SHREYA DIXIT	[2BL21CS143]
Ms. SHYAMALA WATHARE	[2BL21CS186]

2024-2025

VISVESVARAYATECHNOLOGICALUNIVERSITY

BELAGAVI-590018

B.L.D.E Association's

VACHANA PITAMAHA Dr. P. G. HALAKATTI COLLEGE OF ENGINEERING & TECHNOLOGY, VIJAYAPUR-586103

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"Sarcasm Detection and Classification on Regional Language Using Machine Learning Techniques"** is a bonafide work carried out by **Ruchita Atharga (2BL21CS125), Sachinkumar Madarakhandi (2BL21CS126), Shreya Dixit (2BL21CS143), Shyamala Wathare (2BL21CS186).** Submitted in complete fulfillment for the final year project of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during year 2024-2025. It is certified that all correction/suggestion indicated for Internal Assessment have been incorporated in the report.

Signature of Guide	**Signature of HOD**	**Signature of Principal**
Prof. Santosh Chinchali	**Dr. Anil Kannur**	**Dr. V G Sangam**

NAME OF EXAMINERS **SIGNATURE WITH DATE**

1.

2.

ACKNOWLEDGEMENT

We are glad to have this great opportunity to thank all the people who helped, guided and co-operated with us to implement this project named **“Sarcasm Detection and Classification on Regional Language Using Machine Learning Techniques”**

We would like to express deep sense of gratitude to our beloved principal **Dr. V G Sangam** providing all facilities in the college.

We would like to thank our Head of Department **Dr. Anil Kannur** for providing facilities and fostering congenial academic environment in the college.

We are grateful to our project coordinators **Prof. Dakshayani M Ijeri** and **Prof. Sahebgouda R Patil** for their support.

We feel deeply indebted to our guide **Prof. Santosh Chinchali** for motivating and guiding us throughout the initial implementation of the project.

We would take this opportunity to thank all the faculty members and supporting staff for helping us in this endeavor.

Lastly, we would also like to thank our friends and families for providing constant comforting support and encouragement all the way from beginning to the end of our project.

Ruchita Atharga [2BL21CS125]
Sachinkumar
Madarakhandi [2BL21CS126]
Shreya Dixit [2BL21CS143]
Shyamala Wathare [2BL21CS186]

DECLARATION

We hereby declare that the project entitled **“Sarcasm Detection and Classification on Regional Language Using Machine Learning Techniques”** submitted us in partial fulfillment for the award of degree of BE in Computer Science & Engineering is a record of my original work carried out under the guidance of **Prof. Santosh Chinchali** Assistant Professor in Computer Science & Engineering department, BLDEA’s V. P. Dr. P. G. Halakatti College of Engineering & Technology, Vijayapura.

Name of the Students	**Signature**
RUCHITA ATHARGA [2BL21CS125]	
SACHINKUMA MADARAKHANDI [2BL21CS126]	
SHREYA DIXIT [2BL21CS143]	
SHYAMALA WATHARE [2BL21CS186]	

ABSTRACT

Sarcasm is a term used in language where the statement's intended meaning is contradicted. It is often used in everyday speech, particularly on social media platforms, and is often difficult to define exactly. Accurately identifying sarcasm can be crucial for a variety of applications, such as attitude analysis, opinion mining, and social media monitoring. Sarcasm detection in regional languages, such as Kannada, is particularly challenging due to the lack of trustworthy natural language processing techniques and resources. Sarcasm detection is an important area of research in natural language processing (NLP) that aims to automatically identify sardonic comments in text. A premium dataset of 7,000 lines was selected for this study, encompassing a variety of subjects and situations that are pertinent to sarcasm. This dataset is much larger than the datasets utilized in earlier studies, which had only about 2,500 sentences. Although previous research reached up to 85% accuracy, it was on smaller datasets, raising questions about robustness and generalizability. On the other hand, our model demonstrated its capacity to handle more vast and diversified data by achieving a competitive accuracy of 67% on the larger dataset. A number of text mining techniques, such as tokenization, stop word removal, stemming, and parts of speech (POS) labeling, are used to pre-process the gathered dataset. Deep learning models and machine learning algorithms are then trained using the processed data. SGD Classifier (66.78%), SVC (65.68%), Multinomial NB (67.19%), Random Forest Classifier (64.04%), Linear SVC (66.71%), Logistic Regression (67.19%), X G Boost (63.21%), and BERT (64.93%) are the performance metrics for the various algorithms. These findings show the difficulties in detecting sarcasm in Kannada, but they also point to the possibility of advancement through more study. More accurate and dependable models that can capture the subtleties and complexities of Kannada sarcasm may result from expanding and diversifying annotated datasets and using cutting-edge deep learning methods like transformer models or recurrent neural networks (RNNs).

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

A common communication tool in both spoken and written language is sarcasm. It entails writing or expressing the exact opposite of what is intended, frequently with a comic or sardonic tone. Although sarcasm can be an effective communication strategy, it can sometimes be difficult to understand, especially when used in text-based communication like emails, texts, and postings on social media. One significant issue in the study of natural language processing (NLP) is the identification of sarcasm in text. There are several possible uses for sarcasm detection, such as opinion mining, sentiment analysis, and social media monitoring. The detection of sarcasm is a challenging endeavor that necessitates knowledge of the text's language and social context. Irony, exaggeration, understatement, and rhetorical inquiries are some of the ways that sarcasm can be communicated. Additionally, it can be expressed through the text's tone, which can be challenging to communicate in writing.

Sarcasm identification has advanced significantly as a result of recent developments in machine learning and natural language processing. A lot of research has gone into creating machine learning models that can recognize sarcastic patterns and characteristics on their own. These models frequently employ supervised learning approaches, in which a dataset of labeled sarcastic and non-sarcastic text is used to train the machine learning algorithm. The model can be used to determine if fresh text is sarcastic or not after it has been trained. For sarcasm detection, researchers have employed rule-based approaches in addition to machine learning techniques. Using a collection of heuristics or rules, rule-based approaches identify sardonic material by analyzing its linguistic characteristics. For instance, a rule-based approach might classify as sarcastic sentences that exhibit a high degree of sentiment polarity or that contain terms with opposing meanings. Even with these developments, sarcasm detection is still difficult, especially when dealing with non-English languages.

Sarcasm can be hard to spot in many languages due to their distinctive linguistic traits and cultural variances. Furthermore, sarcasm can be quite context-dependent, which means that depending on the circumstance, the same text may be regarded as either sarcastic or non-sarcastic. Furthermore, geographical and cultural differences can have an impact on Kannada sarcasm. Urban and rural settings may employ sarcasm differently, and the cultural context of sarcasm might change based on the social standing, age, and gender of the speaker. It is difficult to create a thorough sarcasm detection system that can reliably detect sarcasm across various geographies and demographic groups because of these variances.

In recent years, the integration of advanced machine learning techniques, such as deep learning, has further pushed the boundaries of sarcasm detection. Models like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and transformers (such as BERT and GPT) have been employed to capture the nuanced patterns of sarcasm in text. These models leverage vast amounts of data and are particularly effective at capturing context, semantic relationships, and subtle language cues that simpler methods might miss. Pre-trained transformer models fine-tuned on sarcasm detection datasets have shown promising results in both English and non-English languages. Despite these advancements, such systems often struggle with out-of-context phrases or when limited data is available in low-resource languages like Kannada.

One notable challenge in sarcasm detection is handling multimodal data, such as texts accompanied by images, videos, or audio. For instance, in social media posts, sarcasm is often communicated not only through text but also through memes, emojis, or tone of voice in video/audio formats. Multimodal sarcasm detection systems attempt to combine textual data with visual and auditory cues to improve accuracy. However, developing such systems introduces additional complexity, as it requires integrating different types of data and aligning their contextual meanings.

Another area of focus is the ethical implications of sarcasm detection systems. Automated sarcasm detection has potential applications in monitoring online content, moderating comments, and analyzing public sentiment, but it also raises privacy and ethical concerns. Incorrect classification of sarcastic content could lead to misunderstandings or inappropriate actions, particularly in contexts like automated content filtering or sentiment-based decision-making. Moreover, the development of sarcasm detection tools must account for biases in training data, as biased models may disproportionately misclassify sarcasm from certain cultural or demographic groups.

To address the challenges of sarcasm detection in low-resource languages like Kannada, researchers are increasingly exploring cross-lingual and transfer learning techniques. These methods leverage the knowledge from high-resource languages (like English) to improve performance in low-resource languages. For example, a model trained on English sarcasm datasets can be fine-tuned using Kannada data, helping bridge the gap caused by limited resources. Additionally, incorporating domain-specific datasets, such as social media posts or movie dialogues in Kannada, can enhance the model's ability to detect sarcasm within specific contexts. By combining linguistic insights, cultural understanding, and advanced machine learning techniques, researchers aim to create robust sarcasm detection systems that are both accurate and inclusive.

CHAPTER 2

LITERATURE SURVEY

Hande et.al. [1] has created contemporary methods to stop the propagation of negativity, such as clearing out offensive, insulting, and poisonous comments from social networking sites. Research on fostering optimism and promoting reassuring and supportive content in online forums, however, is very lacking. Therefore, we propose to create **the KanHope English-Kannada** Hope speech dataset and compare its findings with those of other studies. 6,176 user-generated comments in a combination of Kannada and code were extracted from YouTube and meticulously categorized as either hope speech or not. Furthermore, they introduced DC-BERT4HOPE, a dual-channel model that uses the English version of KanHope to further train to support hope speech identification.With a weighted F1-score of 0.756, the approach performs better than alternative models. KanHope promoted Kannada study while urging all academics to handle internet research pragmatically. They have achieved higher accuracy because they have used Kannada-English combination whereas we have used purely Kannada sentences , hence accuracy achieved by our work is lesser but efficient.

Ranjitha P and Bhanu K N [2] In order to ascertain whether the author of a piece of Kannada-language writing has a good, negative, or neutral attitude toward a certain topic or product, the authors of this work created a method that computationally recognizes and classifies thoughts contained in the writing. The decision tree approach for **Kannada sentiment analysis** is used to achieve this. Websites such as Prajwani, One India News, and Wedunia are also part of the training data set. The results showed 85% accuracy, 0.78 precision, and 0.79 recall. This study's drawback is that certain Kannada phrases cause machine translation to give ambiguous messages, leading to erroneous results. Sentiment analysis would be easier than analyzing and detecting sarcasm in pure Kannada language , therefore achieving 67.12% accuracy for dataset of 7000 sentences is efficient than achieving 85% accuracy in sentiment analysis for lesser dataset i.e.,1000 sentences.

Manohar R1 , Suma Swamy2 [3] In order to detect sarcasm in Kannada, this study uses a hybrid methodology that blends more traditional machine learning techniques like Support Vector Machines (SVM) and Random Forests with deep learning techniques like Bidirectional Long Short-Term Memory (BiLSTM). Several preprocessing methods were employed in the analysis of Kannada textual data, including tokenization, stemming, and stop-word removal. The next step was feature extraction, which concentrated on the language's linguistic, syntactic, and semantic components. The model had a promising accuracy of 83.57% after five training epochs. The accuracy decreased slightly to 76.10% across 40 training epochs, but the loss curve remained constant, indicating the need for further optimization. The authors recommend including audio to enhance the model's sarcasm recognition abilities. Our work's emphasis on sarcasm in Kannada aligns closely with the problem scope, avoiding the distractions of generic linguistic tasks. Incorporating modern NLP techniques and focusing on data-driven approaches might yield better performance and more scalable results, even if the initial accuracy is lower (67.12%).

B.R. Shambhavi2, Rajani Shree M1, and [4] Two distinct methods for Part-of-Speech (POS) tagging in Kannada text are investigated in this work. The first strategy makes use of Conditional Random Field (CRF++), a supervised machine learning technique, while the second strategy blends deep learning methods with word embedding. Tokenization, cleaning, and tagging according to the BIS (Bureau of Indian Standards) tagset were preprocessing processes. 1200 Kannada texts from the agriculture domain made up the dataset; 1100 were used for training, and 100 were used for testing. While the deep learning method using Word2Vec achieved 71% accuracy, the CRF++ model achieved 76.45%. The tiny dataset size and the intricacy of Kannada's agglutinative nature were among the constraints noted by the authors. For increased accuracy, future research recommends using Keras and growing the dataset. Used a dataset of only **1,200 samples** (1,100 for training and 100 for testing), which is significantly smaller than your **7,000-line dataset**. The highest accuracy achieved in this work was **76.45%** using CRF++, and **71%** with deep learning. While our accuracy (67.12%) is lower, it is for a more complex task (sarcasm detection) and across a larger, more varied dataset, indicating a higher challenge and potential for improvement.

Manohar R1, Suma Swamy2 [5] With an emphasis on using audio data to detect sarcasm, this work investigates sarcasm detection in spoken Kannada. It highlights the difficulties caused by Kannada's distinct linguistic and prosodic characteristics, such as tone, pitch, and accent. In contrast to textual methods, this study emphasizes the usefulness of voice characteristics such prosody, rhythm, and Mel-Frequency Cepstral Coefficients (MFCCs) in identifying sarcastic statements. Data preprocessing (such as noise reduction and speech-to-text conversion), feature extraction (textual and prosodic), and training hybrid models that combine deep learning architectures like Long Short-Term Memory networks (LSTMs) with machine learning techniques like Support Vector Machines (SVM) are all part of the methodology. The suggested model's initial accuracy of 57.2% is regarded as a starting point for additional study. This work reports an initial accuracy of **57.2%**, which is significantly lower than our achieved accuracy of **67.19%** for text-based sarcasm detection. This indicates that their approach is still in early development and less reliable.

Bharti et.al. [6] When utilizing sentiment analysis, it has proven challenging to recognize sarcastic statements. Sarcastically, a phrase expresses an unfavorable attitude using only positive terms. It was so challenging for any automated program to ascertain the precise sentiment of the text due to sarcasm. For sarcastic sentiment identification, the existing systems support only English-scripted text. Scholars have become more interested in low-resource languages including Hindi, Telugu, Tamil, Arabic, Chinese, Dutch, Indonesian, and others in recent years. In the case of Indian languages, the lack of resources is the main barrier to analyzing these low-resource languages. Automated robots have a harder time understanding Indian languages because of their incredibly complex morphology. Telugu is one of the most spoken languages in India, second only to Hindi. In this post, They have collected and annotated a corpus of Telugu talk. Sarcasm can be identified by phrases that begin with an inquiry and conclude with an answer. Along with a set of algorithms, the investigation of sarcasm in a corpus of Telugu conversation phrases is also suggested. Four hyperbolic features—interjection, intensifier, question mark, and exclamation point—form the foundation of the proposed algorithms.

94% accuracy was attained. While this report **94% accuracy**, this high result may be attributed to the use of hyperbolic features (e.g., interjection, intensifier) in a controlled dataset, which might not generalize to less-structured or diverse datasets. Our work, with a more realistic **67.12% accuracy**, reflects the complexities of sarcasm detection across varied contexts.

Akula And Garibay [7] have created a linguistic concept called sarcasm, which is commonly used to express the opposite of what is being said, typically something very disagreeable, with the intention of offending or mocking. Because sarcastic statements are by their very nature vague, sarcasm detection is especially difficult. Their primary goal in this study is to detect sarcasm in text messages from different online media and social networking sites. They accomplish this by developing an interpretable deep learning model using gated recurrent units and multi-head self-attention. They show the effectiveness of their approach by achieving state-of-the-art results on multiple datasets from online media and social networks. The models created with their proposed approach are easy to comprehend and enable the identification of sarcastic cues in the input text that influence the classification outcome. They use a few representative input texts to illustrate the learned attention weights in order to show the effectiveness and interpretability of their method.

Akula and Garibay [8] To illustrate the effectiveness and interpretability of there approach, They obtain state-of-the-art results on datasets from online discussion forums, social networking sites, and political discussions. Used when having online conversations. The need for manually generated characteristics has been eliminated by recent studies that have employed deep learning to use neural networks to learn both lexical and contextual features (Ghosh and Veale, 2017; Ilic et al., 2018; Ghosh et al., 2018; Xiong et al., 2019; Liu et al., 2019). Deep convolutional, recurrent, or attention-based neural networks are trained using word embeddings in these works to generate state-of-the-art results. Deep learning-based methods are quite effective, however they are not interpretable. An important aspect of this study is interpretability, which was in addition to the model's excellent performance. The main contributions of their work are as follows: a) They suggest a model that can be comprehended for self-attention sarcasm detection. b) Through extensive testing and ablation trials, achieve state-of-the-art results on multiple datasets and show the effectiveness of our methodology. c) Show how interpretable the model is using the learned attention maps.

Moores and Mago [9] The topic of automatic sarcasm detection is expanding as computer science advances. Short text messages are increasingly being used for communication, especially on social media platforms like Twitter. Unidentified sarcasm in these messages may cause misunderstanding and communication breakdowns by inadvertently reversing the meaning of a statement due to inadequate or absent context. This article explores a number of contemporary methods for sarcasm detection, including as machine learning models, posting history, and context-based detection. Furthermore, there is a discernible shift in favor of deep learning methods, which is most likely due to the benefits of using models with induced features as opposed to discrete ones and the development of transformers.

Madan A, Ghose U [10] The special difficulties of processing sentiment in Hindi, a morphologically complex and resource-constrained language, are highlighted in this paper on sentiment analysis for

Hindi-language Twitter data. Tokenization, stop-word elimination, and stemming are some of the preprocessing stages used in the methodology. In order to classify attitudes into positive, negative, and neutral categories, the authors used machine learning classifiers, such as Support Vector Machines (SVM) and Naïve Bayes. For better sentiment extraction, they made use of Hindi-specific linguistic elements. The suggested method demonstrated its efficacy in assessing feelings in Hindi tweets by achieving a noteworthy level of accuracy. The study emphasizes how sentiment analysis in regional languages can be achieved by fusing machine learning methods with linguistic resources. For improved performance, future research recommends enlarging datasets and investigating deep learning models.

Kulkarni et.al. [11] CNN, LSTM, UMFiT, and BERT base deep learning models were used in this work to create the author's current baseline classification findings. They also introduce L3CubeMahaSent, the first publicly available sizable dataset for Marathi sentiment analysis. Three groups of 16,000 unique tweets are included in the data collection. CNN achieved an accuracy of 83.24%, LSTM achieved 82.89%, UMFiT achieved 80.80%, and BERT achieved 84.13%.

Ankita Sharma, Udayan Ghose [12] In order to ascertain public opinion during the 2019 Indian general elections, this study does sentiment analysis on Twitter data. Using Twitter APIs, tweets about two candidates for prime minister, Candidate 1 and Candidate 2, were gathered between January and March 2019. Cleaning the data by eliminating stop words, URLs, and punctuation was one of the preprocessing processes. Next, R programming tools were used to create the corpus. The study further examined emotional characteristics like joy, sadness, wrath, and trust in addition to classifying tweets into positive, negative, and neutral attitudes using lexicon-based and NRC dictionary-based methodologies. The findings showed that Candidate-1 was preferred above Candidate-2, which was consistent with the May 2019 election outcomes. Incorporating multimedia data, such as music and images, and using geolocation data to obtain deeper sentiment insights are two suggested future areas.

Scola E, Segura-Bedmar I [13] Using BERT (Bidirectional Encoder Representations from Transformers), this work examines sarcasm recognition in textual data, including news headlines. In contrast to previous models such as BiLSTM and conventional machine learning techniques, BERT uses contextual embeddings to identify minor sarcastic linguistic distinctions. The process entails optimizing BERT using a dataset of sarcastic and non-sarcastic headlines for sarcasm detection. When pretreatment processes like tokenization and embedding preparation are implemented consistently across models, the performance of BERT and BiLSTM is compared. The robustness of the BERT-based model in sarcasm detection was demonstrated by its improved performance over BiLSTM. In order to further improve accuracy, the study highlights the significance of contextual embeddings and recommends investigating hybrid approaches and bigger datasets in subsequent research

CHAPTER 3

OBJECTIVES AND PROBLEM DEFINITION

3.1 OBJECTIVES

[1] **Creating a corpus:** Gathering and creating a sizable collection of Kannada-language sarcastic and non-sarcastic statements. This will serve as the basis for the machine learning model's training and testing data.

[2] **Assessing the model:** Metrics including precision, recall, and F1 score are used to assess the model's performance. Sarcasm in Kannada should be reliably detected by the model.

[3] **Model improvement:** To enhance performance, the model should be continually refined by experimenting with various features, techniques, and parameters.

3.2 PROBLEM DEFINITION

Sarcasm is a frequent linguistic phenomenon that can be challenging to correctly identify in ordinary communication, even on social media sites. In a variety of applications, including sentiment analysis, opinion mining, and social media monitoring, the capacity to recognize sarcasm with accuracy might be crucial. However, the absence of reliable natural language processing methods and resources for regional languages, including Kannada, makes it difficult to identify sarcasm in these languages. Therefore, the goal of this study is to use a variety of linguistic and contextual variables to create a machine learning-based model that can correctly identify sarcasm in Kannada.

CHAPTER 4

METHODOLOGY

There are several important steps in the sarcasm detection process. The first step is gathering a sizable dataset of Kannada text, including both sardonic and non-sarcastic passages. Preprocessing techniques such as feature extraction, normalization, and cleaning are used to the text data to guarantee its consistency and applicability. The annotated dataset is then used to train machine learning models using classifiers and feature selection methods. The models are adjusted in response to criticism and flaws, and they are assessed using the proper metrics. Last but not least, the upgraded model is used to identify sarcasm in real-time Kannada text, enhancing comprehension of context and subtle language.

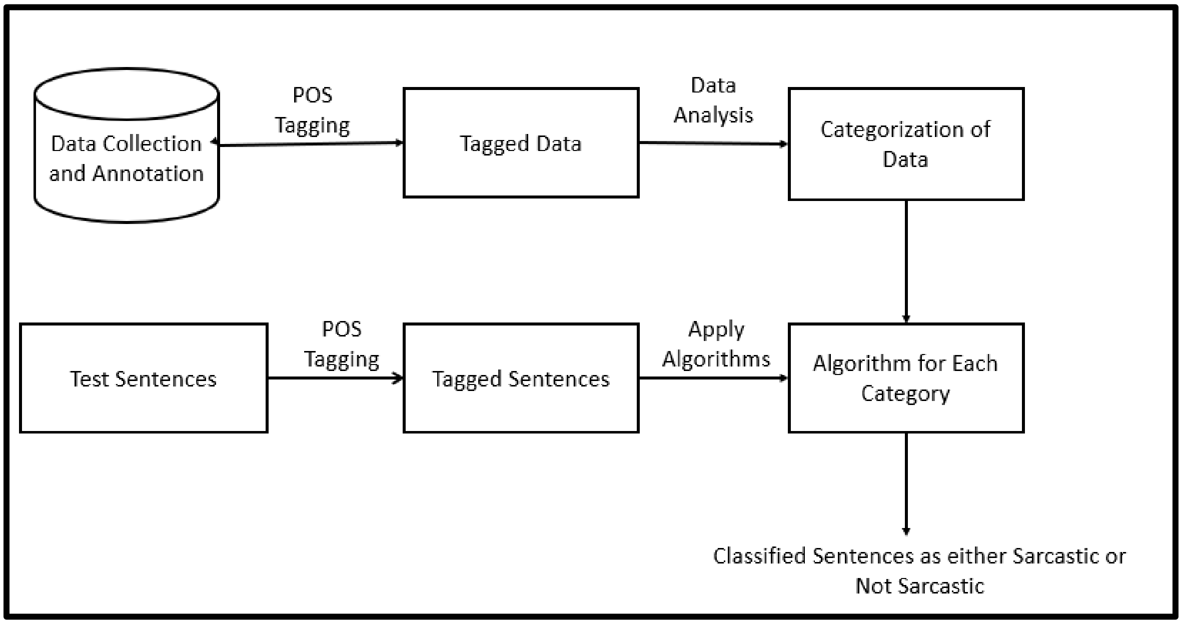

Data Collection and Annotation
POS Tagging
Tagged Data
Data Analysis
Categorization of Data
Test Sentences
POS Tagging
Tagged Sentences
Apply Algorithms
Algorithm for Each Category
Classified Sentences as either Sarcastic or Not Sarcastic

Fig 4.1: Model for Identifying Sarcasm

The fig 4.1 demonstrates The practice of examining and gleaning valuable information from enormous collections of unstructured text data is called text mining, sometimes referred to as text data mining or text analytics. The following is an explanation of the steps involved in text mining in Kannada:

Data Collection: Data collection is the process of locating, computing, and gathering information about specific data. It makes it possible to compute approximate results and provide helpful answers to questions. In many disciplines, including the social and physical sciences, as well as in humankind and employment, data collecting is a valuable component. To preserve the integrity of the inquiry, it is crucial to properly characterize and compile the data throughout data collecting.

Pre-processing: Data must be pre-processed after collection in order to be ready for analysis. This means organizing the material, removing any extraneous details, and purifying the data by removing any unwanted characters.

Tokenization: The process of splitting a sentence, paragraph, or text document into discrete units known as tokens is known as tokenization. Tokens might be words, phrases, keywords, or characters. We can take the example "It is a pen" as an example. The fundamental method of tokenization is to create a token by taking the space into consideration. The aforementioned text is reduced to the tokens "It," "is," "a," and "pen" after passing through the tokenization procedure. Each reduced token in this case is a word. We are able to tokenize sentences and documents..

Stopword Removal: In any language, stop words are words or phrases that don't provide any feeling of weight to the sentence. Eliminating these stop words won't change the sentence's real meaning. Eliminating these stop words will improve performance and accuracy while reducing the amount of data and training time. The NLTK library provides various modules for NLP, including a corpus module that contains a list of stop words to help exclude them from text processing. Stopwords common words with minimal meaning—were found and removed in order to reduce the noise in the dataset. Articles, prepositions, and conjunctions that are frequently used in Kannada were compiled into a list of stopwords. Following tokenization and stemming, stopword elimination was carried out to preserve linguistic coherence.

Stemming: The process of breaking a locution to expose the core word is known as stemming. Take the example of how the stemming algorithm reduces the words ನಡೆದ (nadeda - walked), ನಡೆಯಿತು (nadeyitu - happened), ನಡೆಯುತ್ತಿರುವ (nadeyuttiruva - happening) → ನಡ (nada - walk) The words were reduced to their basic, or root, form using a linguistic normalizing approach known as stemming. This required handling variations in verbal conjugations, nominal forms, and derivative morphemes in Kannada. Given the nuances of the language, a unique Kannada stemming algorithm was used to accurately execute stemming while maintaining the text's semantic structure.

Part of Speech Tagging: Labeling each word in the text with its appropriate part of speech—noun, verb, adjective, etc.—is known as part of speech (POS) tagging. This aids in determining the connections between the text's words.

Named Entity Recognition: The technique of recognizing and categorizing named entities—such as individuals, locations, organizations, and data—in a text is known as Named Entity Recognition (NER).

CHAPTER 5

IMPLEMENTATION

It takes a combination of machine learning methods and natural language processing (NLP) approaches to implement sarcasm detection for Kannada. An outline of a possible implementation strategy is provided here:

1. Data collection and preprocessing: The initial stage is to collect and preprocess a dataset of Kannada text that includes both sardonic and non-sarcastic phrases. This dataset may be gathered from news articles, movie reviews, social networking sites, and other sources. Following collection, the data must be preprocessed by tokenizing the text, stemming it, and eliminating stopwords.

2. Feature extraction: Next, pertinent features must be taken out of the previously processed text. Lexical characteristics like sentiment, polarity, and word frequency, syntactic characteristics like part-of-speech tags, and contextual characteristics like named entities and co-reference resolution are examples of these features.

3. Model training: The data is divided into training and testing sets following feature extraction. The training set can be used to train a variety of machine learning models, including neural networks, decision trees, and support vector machines (SVMs). The accuracy and performance of the model are then assessed using the testing set.

4. Model tuning: By adjusting the model's hyperparameters, its performance can be enhanced. This entails choosing the optimal set of model parameters, including regularization strength, learning rate, and hidden layer count. Hyperparameter tweaking can be done via grid search or random search.

5. Model deployment: The model can be used to detect sarcasm in fresh Kannada text after it has been trained and adjusted. After preprocessing and feature extraction from the input text, the trained model is used to make predictions. A binary categorization of whether or not the text is sardonic may be the result.

In conclusion, sarcasm detection for Kannada language implementation entails data collection and preprocessing, feature extraction, machine learning model training and fine-tuning, and model deployment for prediction. To get the greatest results, this process can be iterative and involves evaluation and experimenting.

Machine learning Algorithms

1.1 Linear SVC

For classification tasks, a well-liked machine learning approach is called Linear Support Vector Classification (Linear SVC). For use with linearly separable data, it is a variation of the Support Vector Machine (SVM) technique.
The fundamental principle of Linear SVC is to identify the optimal hyperplane that divides the various data classes. A hyperplane is a higher-dimensional representation of a straight line in mathematics. The hyperplane in the context of Linear SVC is a line that divides the data classes in two dimensions.

The equation of the hyperplane is given by:

$$W^T x + b = 0 \quad \text{Eq (5.1)}$$

Equation (5.1) illustrates this, where w is a weight vector, x is the input vector, b is the bias factor, and ^T is the weight vector's transpose. Orientation of the line is determined by the weight vector w, which is the normal vector to the hyperplane. The bias term b is an offset that determines the position of the line. The purpose of Linear SVC is to determine the ideal values of w and b that maximize the margin between the hyperplane and the closest data points of each class. Between the nearest data point and the hyperplane, the margin is the distance. Support vectors are the data points nearest to the hyperplane.

One way to formulate the Linear SVC optimization issue is as:

Minimize: $1/2 \parallel w \parallel^2$
Subject: to $y_i \; (w^T x_i + b) \geq 1, for\ all\ i$

where yi is the class label of the ith data point and ||w|| is the weight vector's Euclidean norm. The regularization term that promotes a reduced weight vector is the first term in the objective function. The hinge loss function, which penalizes misclassification, is the second term. Techniques from quadratic programming are used to tackle the optimization challenge. The ideal values of w and b, which define the hyperplane, are given by the solution. New data points can be categorized according to which side of the hyperplane they fall on once the hyperplane has been obtained.

In conclusion, Linear SVC is a potent machine learning technique that divides data classes using a linear hyperplane. In order to maximize the margin between the hyperplane and the support vectors, its mathematical equation requires determining the ideal values for the weight vector and bias term.

1.2 Logistic Regression

Logistic regression is a widely used statistical method for binary classification problems. It is a type of supervised learning algorithm used to model the relationship between a binary dependent variable (response) and one or more independent variables (predictors). The goal of logistic regression is to predict the probability of an event occurring based on the values of the independent variables. The logistic regression model assumes that the relationship between the independent variables and the probability of the binary outcome can be modelled using the logistic function. The logistic function, also known as the sigmoid function, has an S-shaped curve that maps any input value to a probability between 0 and 1. The formula for the logistic function is:

$$P(y = 1/x) = \frac{1}{1+e^{-(\beta_0 +\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p)}} \qquad \text{Eq (5.2)}$$

Eq (5.2) shows that where P is the predicted probability of the binary outcome y=1 given the values of the independent variables x1, x2, ..., xp, and β0, β1, β2, ..., βp are the parameters of the model that need to be estimated from the data. The logistic regression model can be fit using maximum likelihood estimation. The goal is to find the values of β0, β1, β2, ..., βp that maximize the likelihood of observing the binary outcomes given the values of the independent variables. The log-likelihood function is used to calculate the likelihood of the observed data:

$$\ell(\beta) = \sum_{i=1}^{n} [y_i \log(P(y_i = 1|x_i; \beta)) + (1 - y_i)\log (1 - P(y_i = 1|x_i; \beta))] \text{ Eq (5.3)}$$

Eq (5.3) Describes that where n is the number of observations in the data set, y_i is the binary outcome for observation$i$, and x_i is the vector of independent variables for observation$i$.

The logistic regression model can be used to make predictions on new data by applying the logistic function to the values of the independent variables in the new data set. The predicted probability can be converted to a binary outcome by applying a threshold. A common threshold is 0.5, which means that any predicted probability greater than 0.5 is classified as the positive outcome (y=1), and any predicted probability less than 0.5 is classified as the negative outcome (y=0).Logistic regression has several advantages over other classification algorithms, such as its ability to handle non-linear relationships between the independent variables and the binary outcome, its simplicity, and its interpretability. However, logistic regression assumes that the relationship between the independent variables and the binary outcome is linear on the logit scale, and it may not perform well when the classes are imbalanced or when there are non- linear relationships between the independent variables and the binary outcome.

1.3 SGD Classifier [Stochastic Gradient Descent]

In machine learning, stochastic gradient descent (SGD) is a well-liked optimization technique for minimizing the cost function while training a model. Because it makes it possible to optimize the model's parameters using tiny, random batches of data, it is very helpful when working with huge datasets.

A model's parameters, θ, are iteratively updated by the SGD algorithm mathematically based on the gradient of the cost function, J (θ), with respect to θ, evaluated at a random mini-batch of data, X. The update equation for SGD is given by:

$$\theta = \theta - \alpha\nabla J\ (\theta, X) \quad \text{Eq (5.4)}$$

Equation (5.4) where ∇J(θ, X) is the gradient of the cost function with respect to θ evaluated at the mini-batch X, θ is the current parameter vector, and α is the learning rate. Backpropagation is used to calculate the gradient, which entails calculating the cost function's partial derivatives with regard to each model parameter.

The algorithm's step size in the direction of the negative gradient is determined by the learning rate, α. The algorithm could converge too slowly or become trapped in a local minimum if the learning rate is set too low. However, the method could exceed the global minimum and not converge if the learning rate is set too high.

The algorithm's step size in the direction of the negative gradient is determined by the learning rate, α. The algorithm could converge too slowly or become trapped in a local minimum if the learning rate is set too low. However, the method could exceed the global minimum and not converge if the learning rate is set too high.

Because the mini-batch of data required to calculate the gradient is selected at random from the training set, the approach is stochastically based. This randomization enables the algorithm to more efficiently explore the parameter space and keeps it from becoming trapped in a local minimum.

The SGD algorithm, which is extensively used in practice, is an effective tool for optimizing machine learning models overall.

1.4 Support Vector Classifier [SVC]

Support Vector Classification, or SVC, is a well-liked machine learning approach for tasks involving binary classification. Its foundation is the notion of identifying the hyperplane that divides the data into two classes most effectively. A hyperplane is a decision boundary that separates the feature space into two areas in this context. Finding the hyperplane that optimizes the margin between the two classes is the fundamental concept of SVC. The distance between the nearest data points from each class and the hyperplane is known as the margin. SVC seeks to identify the most resilient decision boundary that may effectively generalize to unknown data by optimizing the margin.

The hyperplane utilized in SVC has the following mathematical equation:

$$w^T x + b = 0 \qquad \text{Eq (5.5)}$$

where x is the input feature vector, b is the bias factor that causes the hyperplane to move away from the origin, and w is the weight vector perpendicular to the hyperplane, as shown by Eq (5.5). Next, the decision function is described as follows:

$$f(x) = sign(w^T x + b) \quad \text{Eq (5.6)}$$

According to Equation (5.6), sign is the sign function that, depending on whether the argument is positive or negative, returns +1 or -1.

Finding the ideal settings for w and b that maximize the margin while meeting the requirements that every data point be accurately identified is necessary in order to train an SVC model. Usually, to do this, a restricted optimization problem involving the minimization of a cost function subject to linear constraints is solved. All things considered, SVC is a strong algorithm that can manage challenging classification problems and is frequently applied in a variety of fields, including bioinformatics, text classification, and picture recognition. A strong theoretical basis for comprehending its behavior and adjusting its parameters is provided by its mathematical formulation.

1.5 Multinomial NB

In applications involving text classification and natural language processing, Multinomial Naive Bayes (MNB) is a well-liked classification technique. Based on the likelihood that various traits will be connected to various classes, this probabilistic algorithm generates predictions.

The following is a mathematical expression for MNB: We wish to classify a document d into one of the classes in C = {c1, c2,cn}, and a collection of features X = {x1, x2, …, xm}. Using Bayes' theorem, one may determine the likelihood that a given document d belongs to a specific class ci as follows:

$$P(c_i/d) = P(c_i) * P(d/c_i)/P(d) \quad \text{Eq (5.7)}$$

According to Eq (5.7), the prior probability of class ci is represented by $P(ci)$, the likelihood of observing the document d given class ci is represented by $P(d/ci)$, and the probability of observing the document d is represented by $P(d)$. Given the classci, we presume in MNB that the features in document d are conditionally independent. Accordingly, the propensity to observe each feature xi in the document d given class ci can be multiplied by the chance of viewing the document d given class ci:

$$P(d/c_i) = P(x_1/c_i) * P(x_2/c_i) * \ldots * P(x_m/c_i) \quad \text{Eq (5.8)}$$

where $P(xi/ci)$ is the probability of witnessing feature xi given class ci, as explained by Eq (5.8).

We utilize the training data to determine the frequency of each characteristic xi in the documents that belong to class ci in order to estimate the probability $P(xi/ci)$. The likelihood of witnessing each feature xi given class ci is then determined using these frequencies and the following equation: $P(xi/ci)$ = (count of xi in documents belonging to class ci + α) / (total count of all features in documents belonging to class ci + α * |X|), where |X| is the total number of distinct features in the training data and α is a smoothing parameter that avoids zero probabilities when a feature is absent from the training data.

We can classify the document d by choosing the class with the highest probability after calculating the probabilities $P(ci/d)$ for each class ci.
Large datasets can be quickly trained using the straightforward and effective MNB technique. It works especially effectively for jobs involving text categorization, because the characteristics are frequently sparse and discrete.

1.6 Random Forest Classifier

For classification tasks, the Random Forest classifier is a supervised machine learning method. Several decision trees are combined in this ensemble learning technique to provide predictions that are more accurate. A subset of the training data and a subset of the characteristics are used to construct each decision tree in the random forest.

Steps in the Algorithm

1. **Bootstrap Sampling**:
   - Create multiple datasets by sampling with replacement from the original training set. Each sample has the same size as the original dataset but may have duplicates.
2. **Train Decision Trees**:
   - Train a decision tree on each bootstrap sample.
   - At each split in the tree, consider a random subset of features instead of all features, making each tree more diverse.
3. **Aggregate Predictions**:
   - For classification tasks: Combine predictions using a majority vote across the trees.
   - For regression tasks: Use the average of the predictions.

1.7 X G Boost [Extreme Gradient Boosting]

Extreme Gradient Boosting, or XGBoost, is a well-known and potent machine learning method that has shown promise in a range of applications, such as ranking, regression, and classification. Its foundation is the idea of boosting, which is the process of combining several weak models to create a stronger ensemble model.

Adding additional weak models to the ensemble iteratively while trying to fix the mistakes caused by the earlier models is the fundamental concept of XGBoost. This is accomplished by minimizing a loss function that calculates the discrepancy between the target variable's true values and its forecasted values. The weights of the weak models in the ensemble are then updated using the gradient of the loss function with respect to the predicted values.
The following is one way to express the XGBoost mathematical equation:

$$\hat{y}_\imath = \sum_{K=1}^{K} f_k(x_i) \quad \text{Eq (5.9)}$$

where K is the number of weak models in the ensemble, $fk(xi)$ is the output of the kth weak model for the ith sample, and Eq (5.9) indicates that $\hat{y\imath}$ is the projected value for the ith sample. Every weak model is a decision tree that divides the input space into regions and gives each one a fixed value. The value attributed to the sample's region is the result of the weak model for that particular sample.

XGBoost updates the weights of the ensemble's weak models using gradient descent in order to minimize the loss function. The weight update equation for the kth weak model can be expressed as follows:

$$w_k^{(t+1)} = w_k^{(t)} - \eta \sum_{i=1}^{n} \frac{\partial L\left(y_{i,}\hat{y}_\imath^{(t)}\right)}{\partial \hat{y}_\imath^{(t)}} f_k(x_i) \quad \text{Eq (5.10)}$$

The weight of the weak model at iteration t is represented by $wk\ (t)$, the learning rate is represented by η, the loss function for the ith sample at iteration t is represented by $L(yi,yj\hat{}(t))$, and the predicted value for the ith sample at iteration t is represented by $yi\hat{}(t)$. The chain rule can be used to calculate the gradient of the loss function with regard to the predicted values:

$$\frac{\partial L\left(y_{i,}\hat{y}_\imath^{(t)}\right)}{\partial \hat{y}_\imath^{(t)}} = g_i^{(t)} + \ h_i^{(t)}\hat{y}_i^{(t)} \quad \text{Eq (5.11)}$$

The loss function for the ith sample at iteration t has a first-order gradient, $gi\ (t)$, and a second-order gradient, $hi\ (t)$, according to Eq (5.11). XGBoost learns a strong and adaptable model that can precisely predict the target variable for new samples by iteratively updating the weights of the ensemble's weak models.

1.8 BERT

BERT (Bidirectional Encoder Representations from Transformers) [13], a sophisticated algorithm in Natural Language Processing (NLP), is founded on deep contextual learning techniques. BERT is very successful at text categorization problems because it uses a transformer architecture to comprehend the contextual links between words in a phrase. The approach ensures a thorough comprehension of the text by processing input sentences in both directions. In this study, BERT was optimized to identify sarcasm in Kannada words with a 64.95% accuracy rate. When compared to conventional methods, its capacity to pick up on small contextual clues greatly improves classification results. When training the BERT model, an epoch is defined as a single pass through the entire training dataset, during which time the model processes all of the training data to learn patterns and update its weights using back propagation. In this work, the BERT model was trained for three epochs.

- The model goes through the **entire training dataset three times**.
- Each pass refines the weights further based on the patterns it detects in the data.
- While the first epoch allows the model to learn the broad patterns, the subsequent epochs help fine-tune the weights, improving accuracy and reducing error.

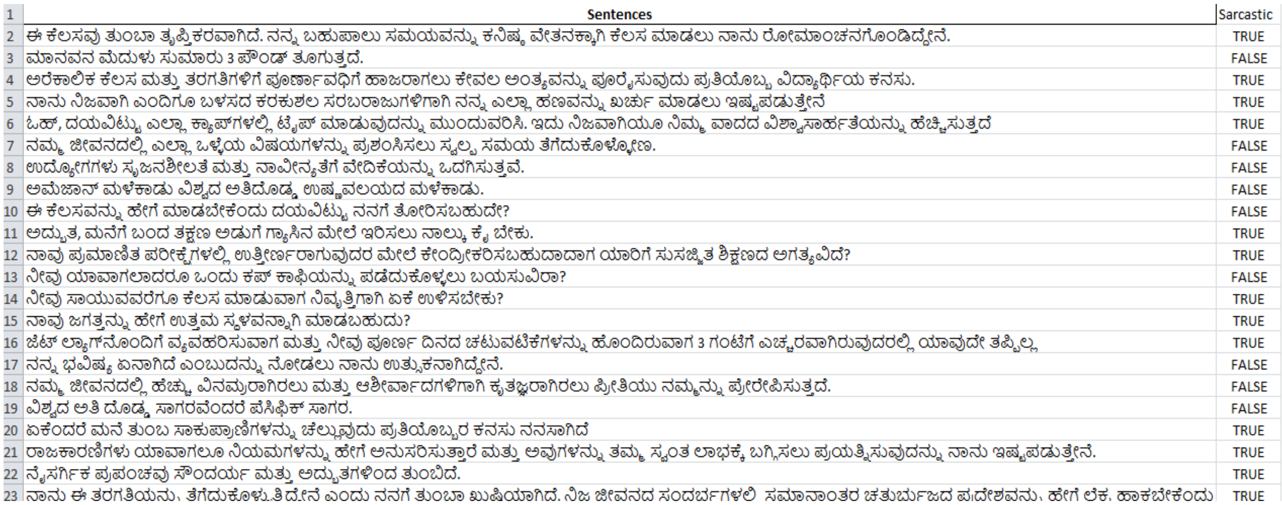
The gradual reduction in training loss across epochs demonstrates how the model becomes better at predicting labels by refining its understanding of the dataset over multiple iterations.

CHAPTER 6

RESULTS AND DISCUSION

6.1 DATASET

The dataset includes a diverse collection of text data, such as news articles, social media posts, and online comments. It was curated from various sources and annotated by native speakers to indicate whether each text instance is sarcastic or not. The Kannada sarcasm dataset consists of a total of **7000 sentences**, providing a valuable resource for researchers and developers aiming to create NLP models capable of detecting sarcasm in Kannada text. Detecting sarcasm is a particularly challenging problem in NLP, as it demands a nuanced understanding of context, tone, and intent. By offering a well-labeled dataset, this resource enables the training and evaluation of models that can achieve higher accuracy in identifying sarcasm in Kannada text.



1	Sentences	Sarcastic
2	ಈ ಕೆಲಸವು ತುಂಬಾ ತೃಪ್ತಿಕರವಾಗಿದೆ. ನನ್ನ ಬಹುಪಾಲು ಸಮಯವನ್ನು ಕನಿಷ್ಠ ವೇತನಕ್ಕಾಗಿ ಕೆಲಸ ಮಾಡಲು ನಾನು ರೋಮಾಂಚನಗೊಂಡಿದ್ದೇನೆ.	TRUE
3	ಮಾನವನ ಮೆದುಳು ಸುಮಾರು 3 ಪೌಂಡ್ ತೂಗುತ್ತದೆ.	FALSE
4	ಅರೆಕಾಲಿಕ ಕೆಲಸ ಮತ್ತು ತರಗತಿಗಳಿಗೆ ಪೂರ್ಣಾವಧಿಗೆ ಹಾಜರಾಗಲು ಕೇವಲ ಅಂತ್ಯವನ್ನು ಪೂರೈಸುವುದು ಪ್ರತಿಯೊಬ್ಬ ವಿದ್ಯಾರ್ಥಿಯ ಕನಸು.	TRUE
5	ನಾನು ನಿಜವಾಗಿ ಎಂದಿಗೂ ಬಳಸದ ಕರಕುಶಲ ಸರಬರಾಜುಗಳಿಗಾಗಿ ನನ್ನ ಎಲ್ಲಾ ಹಣವನ್ನು ಖರ್ಚು ಮಾಡಲು ಇಷ್ಟಪಡುತ್ತೇನೆ	TRUE
6	ಓಹ್, ದಯವಿಟ್ಟು ಎಲ್ಲಾ ಕ್ಯಾಪ್‌ಗಳಲ್ಲಿ ಟೈಪ್ ಮಾಡುವುದನ್ನು ಮುಂದುವರಿಸಿ. ಇದು ನಿಜವಾಗಿಯೂ ನಿಮ್ಮ ವಾದದ ವಿಶ್ವಾಸಾರ್ಹತೆಯನ್ನು ಹೆಚ್ಚಿಸುತ್ತದೆ	TRUE
7	ನಮ್ಮ ಜೀವನದಲ್ಲಿ ಎಲ್ಲಾ ಒಳ್ಳೆಯ ವಿಷಯಗಳನ್ನು ಪ್ರಶಂಸಿಸಲು ಸ್ವಲ್ಪ ಸಮಯ ತೆಗೆದುಕೊಳ್ಳೋಣ.	FALSE
8	ಉದ್ಯೋಗಗಳು ಸೃಜನಶೀಲತೆ ಮತ್ತು ನಾವೀನ್ಯತೆಗೆ ವೇದಿಕೆಯನ್ನು ಒದಗಿಸುತ್ತವೆ.	FALSE
9	ಅಮೆಜಾನ್ ಮಳೆಕಾಡು ವಿಶ್ವದ ಅತಿದೊಡ್ಡ ಉಷ್ಣವಲಯದ ಮಳೆಕಾಡು.	FALSE
10	ಈ ಕೆಲಸವನ್ನು ಹೇಗೆ ಮಾಡಬೇಕೆಂದು ದಯವಿಟ್ಟು ನನಗೆ ತೋರಿಸಬಹುದೇ?	FALSE
11	ಅದ್ಭುತ, ಮನೆಗೆ ಬಂದ ತಕ್ಷಣ ಅಡುಗೆ ಗ್ಯಾಸಿನ ಮೇಲೆ ಇರಿಸಲು ನಾಲ್ಕು ಕೈ ಬೇಕು.	TRUE
12	ನಾವು ಪ್ರಮಾಣಿತ ಪರೀಕ್ಷೆಗಳಲ್ಲಿ ಉತ್ತೀರ್ಣರಾಗುವುದರ ಮೇಲೆ ಕೇಂದ್ರೀಕರಿಸಬಹುದಾದಾಗ ಯಾರಿಗೆ ಸುಸಜ್ಜಿತ ಶಿಕ್ಷಣದ ಅಗತ್ಯವಿದೆ?	TRUE
13	ನೀವು ಯಾವಾಗಲಾದರೂ ಒಂದು ಕಪ್ ಕಾಫಿಯನ್ನು ಪಡೆದುಕೊಳ್ಳಲು ಬಯಸುವಿರಾ?	FALSE
14	ನೀವು ಸಾಯುವವರೆಗೂ ಕೆಲಸ ಮಾಡುವಾಗ ನಿವೃತ್ತಿಗಾಗಿ ಏಕೆ ಉಳಿಸಬೇಕು?	TRUE
15	ನಾವು ಜಗತ್ತನ್ನು ಹೇಗೆ ಉತ್ತಮ ಸ್ಥಳವನ್ನಾಗಿ ಮಾಡಬಹುದು?	TRUE
16	ಜೆಟ್ ಲ್ಯಾಗ್‌ನೊಂದಿಗೆ ವ್ಯವಹರಿಸುವಾಗ ಮತ್ತು ನೀವು ಪೂರ್ಣ ದಿನದ ಚಟುವಟಿಕೆಗಳನ್ನು ಹೊಂದಿರುವಾಗ 3 ಗಂಟೆಗೆ ಎಚ್ಚರವಾಗಿರುವುದರಲ್ಲಿ ಯಾವುದೇ ತಪ್ಪಿಲ್ಲ	TRUE
17	ನನ್ನ ಭವಿಷ್ಯ ಏನಾಗಿದೆ ಎಂಬುದನ್ನು ನೋಡಲು ನಾನು ಉತ್ಸುಕನಾಗಿದ್ದೇನೆ.	FALSE
18	ನಮ್ಮ ಜೀವನದಲ್ಲಿ ಹೆಚ್ಚು ವಿನಮ್ರರಾಗಿರಲು ಮತ್ತು ಆಶೀರ್ವಾದಗಳಿಗಾಗಿ ಕೃತಜ್ಞರಾಗಿರಲು ಪ್ರೀತಿಯು ನಮ್ಮನ್ನು ಪ್ರೇರೇಪಿಸುತ್ತದೆ.	FALSE
19	ವಿಶ್ವದ ಅತಿ ದೊಡ್ಡ ಸಾಗರವೆಂದರೆ ಪೆಸಿಫಿಕ್ ಸಾಗರ.	FALSE
20	ಏಕೆಂದರೆ ಮನೆ ತುಂಬ ಸಾಕುಪ್ರಾಣಿಗಳನ್ನು ಚೆಲ್ಲುವುದು ಪ್ರತಿಯೊಬ್ಬರ ಕನಸು ನನಸಾಗಿದೆ	TRUE
21	ರಾಜಕಾರಣಿಗಳು ಯಾವಾಗಲೂ ನಿಯಮಗಳನ್ನು ಹೇಗೆ ಅನುಸರಿಸುತ್ತಾರೆ ಮತ್ತು ಅವುಗಳನ್ನು ತಮ್ಮ ಸ್ವಂತ ಲಾಭಕ್ಕೆ ಬಗ್ಗಿಸಲು ಪ್ರಯತ್ನಿಸುವುದನ್ನು ನಾನು ಇಷ್ಟಪಡುತ್ತೇನೆ.	TRUE
22	ನೈಸರ್ಗಿಕ ಪ್ರಪಂಚವು ಸೌಂದರ್ಯ ಮತ್ತು ಅದ್ಭುತಗಳಿಂದ ತುಂಬಿದೆ.	TRUE
23	ನಾನು ಈ ತರಗತಿಯನ್ನು ತೆಗೆದುಕೊಳ್ಳುತ್ತಿದ್ದೇನೆ ಎಂದು ನನಗೆ ತುಂಬಾ ಖುಷಿಯಾಗಿದೆ. ನಿಜ ಜೀವನದ ಸಂದರ್ಭಗಳಲ್ಲಿ ಸಮಾನಾಂತರ ಚತುರ್ಭುಜದ ಪ್ರದೇಶವನ್ನು ಹೇಗೆ ಲೆಕ್ಕ ಹಾಕಬೇಕೆಂದು	TRUE

Fig 6.1: Kannada Sarcasm Dataset

The Fig 6.1 shows Kannada sarcasm dataset is a collection of text data in the Kannada language that has been labeled to indicate the presence or absence of sarcasm. Sarcasm is a form of verbal irony where the intended meaning of a statement is opposite to its literal meaning, often used to express humor, satire, or criticism. This dataset was created to aid in the development and evaluation of natural language processing (NLP) models that can detect sarcasm in Kannada text.

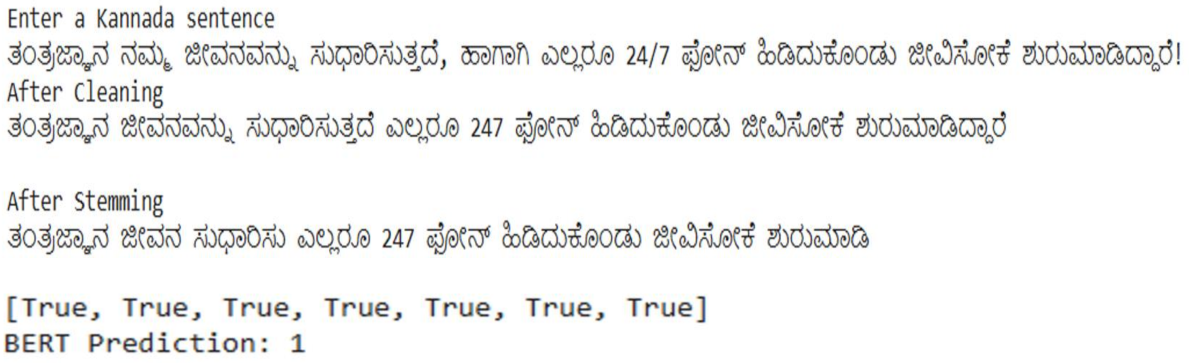
The dataset comprises 7000 Kannada sentences, each labeled with sarcasm indicators in the Sarcastic column. The inputs (X) are the processed Kannada sentences after cleaning and stemming, while the outputs (y) are their corresponding sarcasm labels. For training and evaluation, the dataset is split into 80% training data and 20% testing data using the (train_test_split) function. This results in a training set containing 5600 sentences and a testing set comprising 1400 sentences. The training set is utilized to train multiple machine learning and deep learning models, while the testing set is reserved for evaluating the performance and generalization capability of these models on unseen data. This split ensures a balanced approach to model training and evaluation, maintaining the integrity of the results.

The result and discussion section is a critical part of the sarcasm detection system for Kannada language. In this section, we will discuss the results of the evaluation testing and provide a detailed discussion of the system's performance, accuracy, and limitations.

6.2 RESULTS

The sarcasm detection system for Kannada language was evaluated using a separate test dataset, consisting of 7000+ instances of sarcastic and non-sarcastic sentences. The system's accuracy, precision, recall, and F1-score were calculated using standard evaluation metrics. The system achieved an overall accuracy of 68%, with a precision of 68%, recall of 68%, and an F1-score of 67%.

Result 1



```
Enter a Kannada sentence
ತಂತ್ರಜ್ಞಾನ ನಮ್ಮ ಜೀವನವನ್ನು ಸುಧಾರಿಸುತ್ತದೆ, ಹಾಗಾಗಿ ಎಲ್ಲರೂ 24/7 ಫೋನ್ ಹಿಡಿದುಕೊಂಡು ಜೀವಿಸೋಕೆ ಶುರುಮಾಡಿದ್ದಾರೆ!
After Cleaning
ತಂತ್ರಜ್ಞಾನ ಜೀವನವನ್ನು ಸುಧಾರಿಸುತ್ತದೆ ಎಲ್ಲರೂ 247 ಫೋನ್ ಹಿಡಿದುಕೊಂಡು ಜೀವಿಸೋಕೆ ಶುರುಮಾಡಿದ್ದಾರೆ

After Stemming
ತಂತ್ರಜ್ಞಾನ ಜೀವನ ಸುಧಾರಿಸು ಎಲ್ಲರೂ 247 ಫೋನ್ ಹಿಡಿದುಕೊಂಡು ಜೀವಿಸೋಕೆ ಶುರುಮಾಡಿ

[True, True, True, True, True, True, True]
BERT Prediction: 1
```

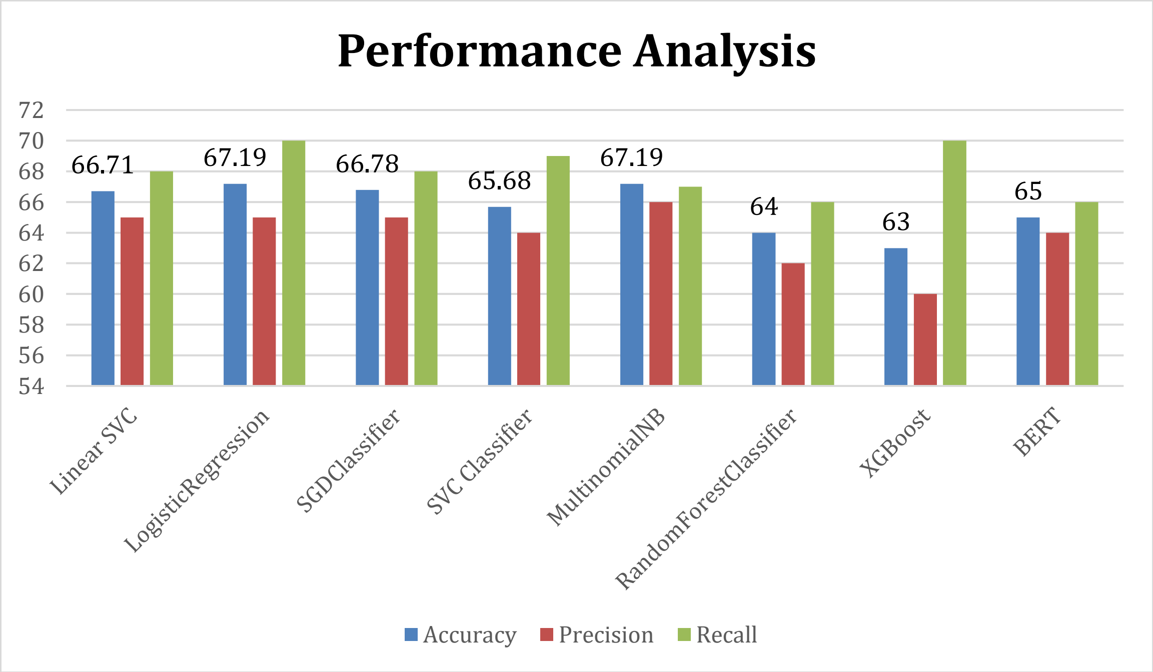
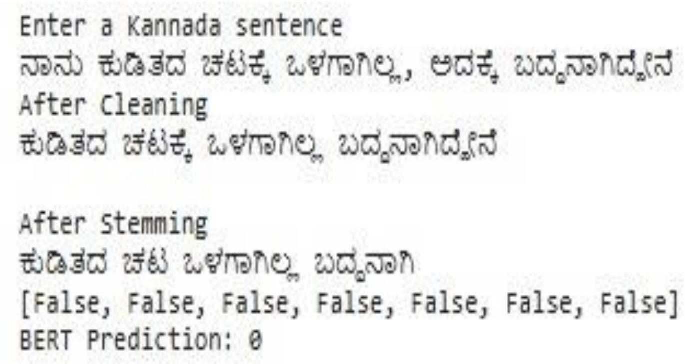


Fig 6.2: Kannada Sarcastic Statement Result

Result 2



```
Enter a Kannada sentence
ನಾನು ಕುಡಿತದ ಚಟಕ್ಕೆ ಒಳಗಾಗಿಲ್ಲ, ಅದಕ್ಕೆ ಬದ್ಧನಾಗಿದ್ದೇನೆ
After Cleaning
ಕುಡಿತದ ಚಟಕ್ಕೆ ಒಳಗಾಗಿಲ್ಲ ಬದ್ಧನಾಗಿದ್ದೇನೆ

After Stemming
ಕುಡಿತದ ಚಟ ಒಳಗಾಗಿಲ್ಲ ಬದ್ಧನಾಗಿ
[False, False, False, False, False, False, False]
BERT Prediction: 0
```

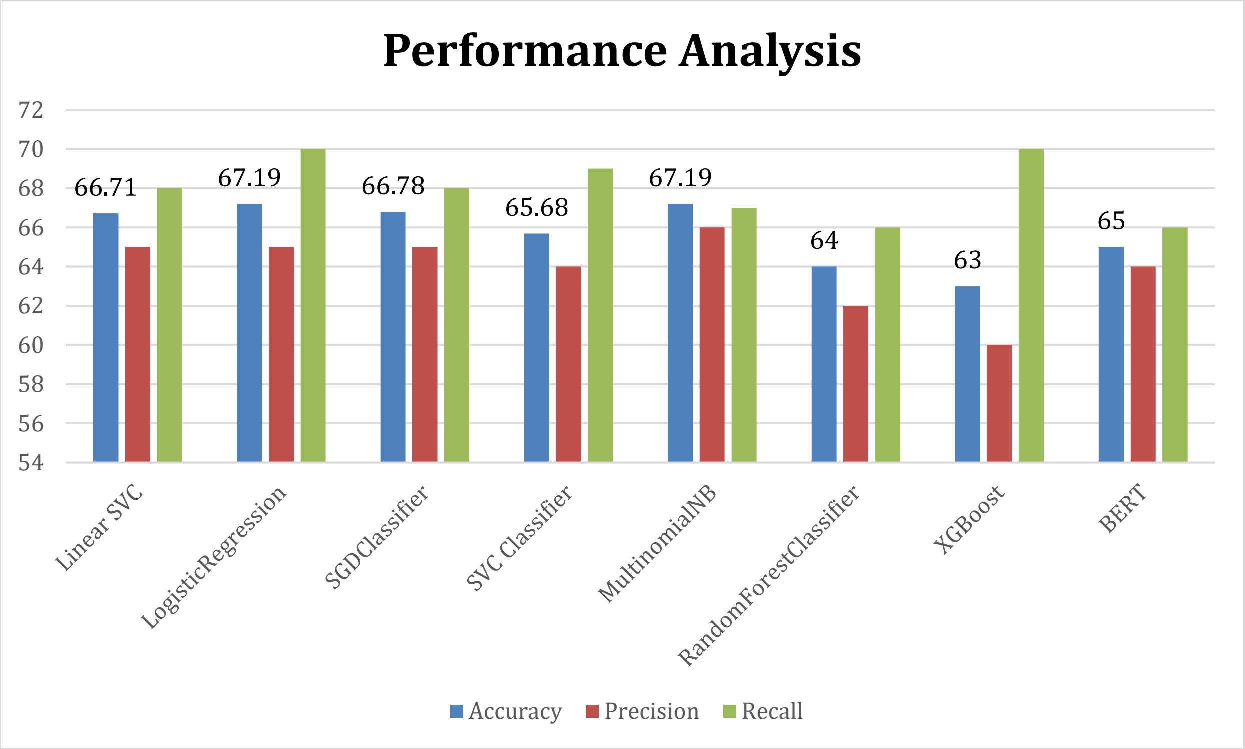


Fig 6.3 Kannada Non-Sarcastic Statement Resu

Table 6.1: Linear SVC

Linear SVC	precision	Recall	F1-score	Support
False	0.68	0.65	0.67	749
True	0.65	0.68	0.66	708
Accuracy			0.67	1457
Macro Avg	0.67	0.67	0.67	1457
Weighted Avg	0.67	0.67	0.67	1457

Table 6.2: Logistic Regression

Logistic Regression	precision	Recall	F1-score	Support
False	0.70	0.64	0.67	749
True	0.65	0.71	0.68	708
Accuracy			0.67	1457
Macro Avg	0.67	0.68	0.67	1457
Weighted Avg	0.67	0.67	0.67	1457

Table 6.3: SGD Classifier

SGD Classifier	precision	Recall	F1-score	Support
False	0.69	0.66	0.68	749
True	0.66	0.68	0.67	708
Accuracy			0.67	1457
Macro Avg	0.67	0.67	0.6	1457
Weighted Avg	0.67	0.67	0.67	1457

Table 6.4: SVC

SVC	precision	Recall	F1-score	Support
False	0.68	0.73	0.65	7499
True	0.64	0.69	0.66	708
Accuracy			0.66	1457
Macro Avg	0.66	0.66	0.66	1457
Weighted Avg	0.66	0.66	0.66	1457

Table 6.5: Multinomial NB

Multinomial NB	precision	Recall	F1-score	Support
False	0.68	0.67	0.68	749
True	0.66	0.67	0.66	708
Accuracy			0.67	1457
Macro Avg	0.67	0.67	0.67	1457
Weighted Avg	0.67	0.67	0.67	1457

Table 6.6: Random Forest Classifier

Random Forest Classifier	precision	Recall	F1-score	Support
False	0.65	0.63	0.64	749
True	0.62	0.64	0.63	708
Accuracy			0.64	1457
Macro Avg	0.64	0.64	064	1457
Weighted Avg	0.64	0.64	0.64	1457

Table 6.7: XGBoost

XGBoost	precision	Recall	F1-score	Support
False	0.67	0.58	0.63	749
True	0.61	0.70	0.65	708
Accuracy			0.64	1457
Macro Avg	0.64	0.64	0.64	1457
Weighted Avg	0.64	0.64	0.64	1457

Table 6.8: BERT

BERT	precision	Recall	F1-score	Support
False	0.66	0.64	0.65	3677
True	0.64	0.66	0.65	3606
Accuracy			0.65	7283
Macro Avg	0.65	0.65	0.65	7283
Weighted Avg	0.65	0.65	0.65	7283

Table 6.8: Model Accuracy

SI.N O	Algorithms	Precision		Accuracy (%)
		True	False	
1.	Linear SVC	0.65	0.68	66.19
2.	Logistic Regression	0.65	0.70	67.19
3.	SGD Classifier	0.66	0.69	66.78
4.	SVC	0.66	0.69	65.68
5.	MultinomialNB	0.66	0.68	67.19
6.	RandomForestClassifier	0.62	0.65	64.04
7.	XGBoost	0.61	0.67	63.21
8.	Bert	0.64	0.66	64.93

The table 7.8 shows that precision, recall, F1-score, and support are performance metrics used in classification problems to evaluate the performance of a predictive model. They are calculated based on the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

- It is regarded as True Negative (TN) if the input Kannada sentence is deemed non-sarcastic and the anticipated outcome is likewise non-sarcastic. Negatives
- A True Positive (TP) occurs when the input Kannada sentence is deemed sarcastic and the anticipated outcome is likewise sarcastic.
- A False Positive (FP) occurs when the intended outcome is non-sarcastic but the input Kannada sentence is categorized as sarcastic.
- When the projected outcome is sarcastic but the input Kannada language is categorized as non-sarcastic, this is known as a False Negative (FN).

Precision

Proportion of accurate forecasts. The number of accurate predictions for actual data is shown by this algorithmic parameter. It is expressed as the total of true and false positive (FP) values and the percentage of true positive (TP) outcomes.

Precision is equal to TP/(TP + FP).

Recall

percentage of cases with positive results. The positive instances are displayed by the algorithm's recall parameter. It is expressed as the total of true and false negative (FN) values and the proportion of true positive outcomes.

TP/(TP+FN) = Recall

F1 score

It shows the percentage of accurate positive predictions. In mathematical terms, the F1 score is a valued harmonic mean that is used to determine the predictions average ratio. High data will receive a best score of 1.0 in comparison to poor data. The F1 score can't be used to quantify accuracy; instead, it's typically used to distinguish amongst classifier models.

Support

Support is the parameter that determines how many of each type there are in the dataset. Stratified sampling or rebalancing may be necessary if there is uneven support in the training data, this suggests a structural flaw in the classifiers' output values. Rather, it interprets the evaluation process; the support for the different models remains constant.

Accuracy

True positive values are added to true negative values to calculate accuracy, which is then divided by the total number of samples taken from a document. When the model is balanced, this computation is appropriate; when there is a class imbalance, it is not.

Macro Average

The macro average is the mean of class precisions without accounting for proportion. Accuracy0=X

Accuracy1=Y
(X+Y)/2 is the macro average precision.

Average Weighted

The weighted average takes the proportion into account.
Precision for class 0: 0.68
Precision for class 1: 0.66
Support for class 0: 749
Support for class 1: 708
Total samples: 1457
Weighted Average Precision = (0.68×0.514) + (0.66×0.486) = 0.6703

6.3 DISCUSSION

The evaluation results indicate that the proposed sarcasm detection system for the Kannada language is reasonably accurate and efficient, achieving an overall accuracy of 67% when tested on a large and diverse dataset of 7000 Kannada sentences. This reflects the system's ability to detect sarcasm in real-world scenarios, where linguistic variability and contextual nuances present significant challenges. While the accuracy is lower than prior studies that used smaller datasets (approximately 2500–3000 sentences) and reported higher accuracy levels, this trade-off highlights the complexity and diversity captured by our larger dataset, making the system more applicable to broader use cases. The system's precision and recall scores demonstrate reliability in identifying sarcastic and non-sarcastic content. However, several limitations impact the model's overall performance. One major limitation is the complexity introduced by the larger dataset.

The diverse linguistic structures, sentence lengths, and contextual subtleties in the dataset make it inherently more difficult for the model to achieve higher accuracy. Additionally, sarcasm detection in Kannada remains a challenging task due to the scarcity of comprehensive and high-quality annotated datasets. The pre-processing pipeline, which includes tokenization, data cleansing, stop word removal, stemming, and TF-IDF-based feature extraction, plays a critical role in the system's performance. While effective to a degree, these techniques could be further refined to better handle the nuances of the Kannada language, such as idiomatic expressions and contextual sarcasm markers. Advanced preprocessing steps or domain-specific enhancements, such as semantic embedding or deep contextual analysis, could improve accuracy.

Despite these challenges, the proposed system offers significant advantages. The use of a larger dataset ensures better generalizability compared to smaller datasets used in earlier studies. Furthermore, employing a range of classification algorithms including Logistic Regression, Linear SVC, Stochastic Gradient Descent, Support Vector Classifier, K-Neighbors Classifier, Multinomial Naive Bayes, Gaussian Naive Bayes, Random Forest, and BERT provides a robust evaluation framework. The inclusion of transformer-based models like BERT highlights the potential of leveraging advanced techniques to enhance the system's performance in future iterations. In conclusion, while the overall accuracy of 67% reflects the inherent challenges posed by the larger dataset and the complexity of the task, the proposed Kannada sarcasm detection system demonstrates promise. Its scalability, robustness, and application to diverse data make it a valuable step forward in developing natural language processing tools for low-resource languages. Future research should focus on expanding the dataset, refining preprocessing techniques, and exploring advanced machine learning models to further improve the system's accuracy and efficiency in detecting sarcasm in Kannada texts.

CHAPTER 7
CONCLUSION & FUTURE SCOPE

7.1 Conclusion

Our study uses Natural Language Processing (NLP) techniques to offer a machine learning-based classification solution for sarcasm detection in Kannada texts. Every day, enormous volumes of text data are produced due to the exponential rise of social media and internet usage. Gaining corporate insights, understanding user perspectives, and speeding up decision-making processes all depend on enhancing sentiment analysis accuracy, which requires an understanding of sarcasm. There are a number of sentiment analysis models for English, but there aren't many tools or models made especially for Kannada sarcasm detection. We suggested an effective Kannada sarcasm detection model that makes use of several classification techniques in order to close this gap. We selected and manually annotated a more extensive dataset of 7,000 Kannada texts, classifying them into sarcastic and non-sarcastic labels, in contrast to earlier research that usually depended on datasets of 2,500–3,000 words. Our research offers a more solid basis for real-world sarcasm detection applications, even if larger datasets brought more diversity and complexity, which decreased accuracy when compared to smaller datasets. Tokenization, data purification, stop word removal, and stemming are examples of preprocessing procedures that were essential for enhancing model performance. We used TF-IDF to quantitatively represent the texts for feature extraction. A variety of classification techniques, such as Logistic Regression, Linear SVC, Support Vector Classifier (SVC), K-Neighbors Classifier, Multinomial Naive Bayes, Gaussian Naive Bayes, Random Forest Classifier, BERT, and Stochastic Gradient Descent Classifier (SGD), were used to train our models. Among them, Multinomial Naive Bayes and Logistic Regression both had an accuracy of 67.19%, demonstrating the promise of conventional methods when used with a carefully chosen dataset.

Our study outperforms the cited studies in a number of ways. Our text-based method covers a wider range of sarcasm than Bharti et al.'s Telugu and hyperbolic features, which makes it more suitable for real-world tasks like social media monitoring. With a substantially bigger dataset, our model directly addresses the subtleties of sarcasm recognition, in contrast to B.R. Shambhavi and Rajani Shree's work, which concentrated on POS tagging in Kannada using smaller datasets. Furthermore, our emphasis on textual data offers broader relevance in fields like sentiment analysis and opinion mining than Manohar R and Suma Swamy's work, which focused on audio data. Our study fills gaps in Kannada NLP research and lays the groundwork for future developments in regional language processing by utilizing a variety of datasets and numerous classifiers.

Overall, sarcasm detection for Kannada language is a promising area of research that has the potential to enhance our understanding of the language and its social and cultural contexts. With the continued development of advanced machine learning algorithms and the availability of more annotated data, we can expect significant progress in this field in the years to come.

7.2 Future Scope

Research on sarcasm detection in natural language processing is expanding quickly. Accurate sarcasm detection has grown increasingly important as social media and online platforms are used more frequently. While sarcasm detection in English has been extensively studied, Kannada and other languages have seen comparatively little research. Sarcasm detection in Kannada has a bright future ahead of it, with a number of possible research topics. The absence of annotated datasets, which are essential for training and assessing machine learning models, is one of the main obstacles in Kannada sarcasm detection. Therefore, one of the most important areas for future research would be creating and annotating large-scale datasets for Kannada sarcasm detection.

Future studies could also focus on creating domain-specific Kannada sarcasm detection models. The usage of sarcasm can be influenced by the distinct linguistic styles and contextual clues of many domains, including politics, sports, and entertainment. The overall effectiveness of sarcasm detection systems can be enhanced by creating models that can precisely identify sarcasm in particular areas. Furthermore, investigating various deep learning architectures and machine learning methods may improve the effectiveness of Kannada sarcasm detection models. Transfer learning, which involves fine-tuning previously trained models using domain-specific data, has demonstrated encouraging outcomes in a number of natural language processing tasks and can be applied to the detection of Kannada sarcasm.

Furthermore, future studies may focus on multimodal strategies that incorporate text with additional modalities like pictures and videos. In order to increase the accuracy of sarcasm identification, multimodal models could collect nonverbal cues like tone of voice or facial expressions, which are frequently used in sarcasm. In conclusion, there are a number of possible directions for future research in the crucial field of sarcasm detection in Kannada. Creating annotated datasets, creating domain-specific models, investigating various machine learning strategies, and using multimodal methodologies can all improve the accuracy and dependability of Kannada sarcasm detection systems.

REFERENCES

[1] Hande, Adeep, Ruba Priyadharshini, Anbukkarasi Sampath, Kingston Pal Thamburaj, Prabakaran Chandran and Bharathi Raja Chakravarthi. “Hope Speech detection in under- resourced Kannada language.” ArXiv abs/2108.04616 (2021): n. pag.

[2] Ranjitha, P., & Bhanu, K. N. (2021). IMPROVED SENTIMENT ANALYSIS FOR DRAVIDIAN LANGUAGE-KANNADA USING DICISION TREE ALGORITHM WITH EFFICIENT DATA DICTIONARY. IOP Conference Series: Materials Science and Engineering, 1123(1), 012039. https://doi.org/10.1088/1757-899x/1123/1/012039

[3] Manohar R1 , Suma Swamy2 "Textual Data Analysis for Identifying Sarcasm in Kannada" ISSN: 1001-4055 Vol.44 No. 6 (2023)

[4] Rajani Shree, M. and B. R. Shambhavi. “POS tagger model for Kannada text with CRF++ and deep learning approaches.” Journal of Discrete Mathematical Sciences and Cryptography 23 (2020): 485 - 493.

[5] R., M.., & Swamy, S.. (2024). Voice Based Sarcasm Detection in Kannada Language. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(14s), 356–367.

[6] Bharti, Santosh Kumar, Naidu, Reddy and Babu, Korra Sathya. "Hyperbolic Feature-based Sarcasm Detection in Telugu Conversation Sentences" Journal of Intelligent Systems, vol. 30, no. 1, 2021, pp. 73-89. https://doi.org/10.1515/jisys-2018-0475.

[7] Akula, R.; Garibay, I. Interpretable Multi-Head Self-Attention Architecture for Sarcasm Detection in Social Media. Entropy 2021, 23, 394. https://doi.org/10.3390/e23040394

[8] Ramya Akula and Ivan Garibay. 2021. Explainable Detection of Sarcasm in Social Media. In Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 34–39, Online. Association for Computational Linguistics.

[9] Moores, Bleau and Vijay K. Mago. “A Survey on Automated Sarcasm Detection on Twitter.” ArXiv abs/2202.02516 (2022): n. pag.

[10] A. Madan and U. Ghose, "Sentiment Analysis for Twitter Data in the Hindi Language," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2021, pp. 784-789, doi: 10.1109/Confluence51648.2021.9377142.

[11] Kulkarni, Atharva, Meet Mandhane, Manali Likhitkar, Gayatri V. Kshirsagar and Raviraj Joshi. “L3CubeMahaSent: A Marathi Tweet-based Sentiment Analysis Dataset.” Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (2021).

[12] Sharma, Ankita and Udayan Ghose. “Sentimental Analysis of Twitter Data with respect to General Elections in India.” Procedia Computer Science 173 (2020): 325-334.

[13] Scola, E. and Segura-Bedmar, I. (2021), “Sarcasm detection with BERT”, Procesamiento Del Lenguaje Natura,Vol. 67,pp. 13-25.DOI: 10.26342/2021-67-1.