REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

SARCASM DETECTION FROM TEXT WITH CONTEXT INFORMATION USING DEEP LEARNING

Usman MUHAMMAD

Master`s Thesis

Supervisor

Asst. Prof. Dr. Abdullahı Abdu IBRAHIM

Istanbul, 2023

SARCASM DETECTION FROM TEXT WITH CONTEXT INFORMATION USING DEEP LEARNING

Usman MUHAMMAD

Electrical and Computer Engineering

Master`s Thesis

ALTINBAŞ UNIVERSITY

2023

The thesis titled SARCASM DETECTION FROM TEXT WITH CONTEXT INFORMATION USING DEEP LEARNING prepared by USMAN MUHAMMAD and submitted on 24/04/2023 has been **accepted unanimously** for the degree of Master Electrical and Computer Engineering.

Asst. Prof.Dr.Abdullahı Abdu IBRAHIM

Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Abdullahı Abdu IBRAHIM	Department of Computer Engineering, Altınbaş University	
Asst. Prof. Dr. Timur İNAN	Department of Software Engineering, Altınbaş University	
Asst. Prof. Dr. Tareq Abed MOHAMMED	Department of Information Technology, Kirkuk University	

I hereby declare that this thesis meets all format and submission requirements of a Master`s Thesis.

Submission date of the thesis to the Institute of Graduate Studies: ____/____/____

I hereby declare that all information presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Usman MUHAMMAD

Signature

DEDICATION

I would want to thank Allah first for allowing me to conquer every challenge. I wish to thank my supervisor (Asst. Prof Dr. Abdullahi Abdu IBRAHIM) for his guidance and support during my academic career. I received his guidance and support at every level of my project's development. I want to thank the members of my committee for letting my defence be entertaining and for their wise comments and recommendations.

PREFACE

I must extend my thanks and appreciation to my master’s thesis supervisor, (Asst. Prof Dr. Abdullahi Abdu IBRAHIM), for his directives and instructions. He was very helpful to me, and I learned a lot from him. A special thank you for his sincerity, and I wish a happy life for him.

ABSTRACT

SARCASM DETECTION FROM TEXT WITH CONTEXT INFORMATION USING DEEP LEARNING

MUHAMMAD, Usman

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof Dr. Abdullahi Abdu IBRAHIM

Date: April / 2023

Pages: 69

Deep learning methods have received a great deal of attention recently and have advanced remarkably in the field of natural language processing. The subjective and context-dependent character of sarcasm makes it a difficult challenge to solve, making it one of the essential objectives in this discipline. Sarcasm is a typical human behavior that is frequently used to transmit, mock, or criticize in a humorous way when one has an unfavorable view. Because sarcastic phrases frequently have different meanings from genuine ones, they are different from ordinary conversation. Many studies are being conducted on various social networks and social media platforms as a result of the increase in the volume of opinionated content on these platforms over time. Sarcasm plays a big part in social media networks since it is so common in tweets and comments. At this point, the focus is on contemporary, application-oriented, fine-grained aspects of the user-generated text, such as posture, emotions, and attitude. To identify the authors' true intents, numerous previous models have been developed to detect sarcasm on the principles of syntactical, lexical, and semantic analysis of user-generated content. Using context data and deep learning techniques, we offer a unique method for sarcasm identification from the text in this study. In this study, lexical, pragmatic, and semantic variables are also used to identify sarcasm. We put out a deep neural network-based contextual sarcasm sensor that can recognize sarcasm from context. A contextual sarcasm detector incorporates user embedding, which records the user's personality and its stylometric characteristics because the manner of expression and sarcasm's inherent variability differ from person to person. This suggested model has a 91% accuracy rate. It measured the performance impact of transfer learning for classifying Urdu documents with 72 percent accuracy using the BERT classifier. CNN and Word2Vec have both been used to extract content-based characteristics from embeddings.

Keywords: Sarcasm Detection, Deep Learning, Social Data, Convolutional Neural Network, BERT

TABLE OF CONTENTS

Pages

ABSTRACT vii

LIST OF TABLES xii

LIST OF FIGURES xiii

ABBREVIATIONS xiv

1. INTRODUCTION 1

1.1 BACKGROUND 1

1.2 SARCASM DETECTION ADVANTAGES 4

1.2.1 Sentiment Analysis 5

1.2.2 Customer Service 6

1.2.3 Brand Reputation 6

1.2.4 Social Media Monitoring 7

1.3 SARCASM DETECTION IMPLICATIONS 9

1.4 THESIS OBJECTIVES 9

1.5 MOTIVATION 10

1.6 PROBLEM STATEMENT 11

1.7 RESEARCH SIGNIFICANCE 11

1.8 THESIS STRUCTURE 12

2. LITERATURE REVIEW 13

2.1 LINGUISTIC THEORIES OF SARCASM DETECTION 13

2.1.1 Irony Markers 13

2.1.2 Model Verbs 14

2.1.3 Polarity Cues 14

2.1.4 Ambiguity 14

2.2 AUTOMATIC SARCASM DETECTION APPROACHES 15

2.2.1 Rule-Based System 15

2.2.2 Machine Learning-Based Systems 16

2.2.3 Deep Learning-Based systems 18

2.3 RELATED WORK 20

2.3.1 Machine Learning Approach 20

2.3.2 Content Feature-Based Analyses 21

2.3.3 Context-Based Models 22

2.3.4 Pattern-Based Methods 24

2.3.5 Automated Machine Learning (AutoML) 27

2.3.6 Deep Learning Approach 32

3. METHODOLOGY 37

3.1 ADOPTED METHODOLOGIES FOR SARCASM DETECTION 37

3.1.1 Dataset Description 37

3.1.2 Dataset Preprocessing 38

3.1.3 Proposed Model Architecture 38

4. EXPERIMENTS AND RESULTS 41

4.1 EXPERIMENTAL SETUP 41

4.2 MODEL RESULTS 42

4.3 CONFUSION MATRIX PARAMETERS 44

4.4 ACCURACY 45

4.5 GUI 47

5. DISCUSSION AND CONCLUSION 49

5.1 CONCLUSION 49

5.2 DISCUSSION 50

5.3 FUTURE WORK 52

REFERENCES 54

LIST OF TABLES

<u>Pages</u>

Table 2.1: Summarization of Literature Review 35

Table 3.1: Dataset Description 38

Table 4.1: Results of Model 43

LIST OF FIGURES

Pages

Figure 1.1: Tweet Cloud 2

Figure 1.2: Social Media Monitoring 8

Figure 3.1: Adopted Methodology Architecture 37

Figure 3.2: Softmax View 39

Figure 3.3: BERT Model Architecture 40

Figure 4.1: Confusion Matrix 45

Figure 4.2: Accuracy Graph 47

Figure 4.3: GUI Screen of Our Proposed System. 48

ABBREVIATIONS

BERT	:	Bidirectional Encoder Representations from Transformers
TF-IDF	:	Term Frequency-Inverse Document Frequency
DL	:	Deep Learning
SVM	:	Support Vector Machine
AI	:	Artificial Intelligence
POS	:	Point of Sales
SMO	:	Sequential Minimal Optimization
LSTM	:	Long Short-Term Memory Networks
LIBSVM	:	Library for Support Vector Machines
CNN	:	Convolutional Neural Network
QP	:	Quadratic Programming
CV	:	Cross-Validation
NLP	:	Natural language processing
DT	:	Decision Tree
Word	:	Words Into Numerical Vectors

1. INTRODUCTION

1.1 BACKGROUND

Sarcasm plays a vital role in human conversation [4]. As it helps people not to pretend but express their true feelings, intentions, and emotions. Just for sake of an example, words and wordless actions are also used for the representation of sarcasm, such as a change of tone, the stress laid on words while speaking, and some stunning or feared facial expressions. Mainly sarcasm requires additional contextual knowledge to understand what the speaker is trying to say. To identify the neural correlates of sarcasm, neuropsychology experts have dissected different patterns of brain activity. They point out that understanding sarcasm depends a lot on the speaker's personality, emotional state, as well as facial expressions and prosody [1]. It also depends a lot on the context of an utterance. Our conversation turns out to be interesting with a sarcastic factor, but the complexity of this task was the thing that researchers were refraining from it. Naturally, sarcasm is an expression of personal sentiments which are contrary to implied ones [2]. Words used in sarcastic sentences are often opposite in their meanings [3] because you want to manipulate meanings that are residing inside that word. The proposed model extracts the features from the dataset mean keywords context from the data and then classifies the dataset into sarcastic and non-sarcastic. The keyword cloud is shown in figure 1.1. The main agenda of sarcasm includes humiliation, expressing irritation, or presenting something as fun. Detection of sarcasm is important for sentimental analysis and other research firms.

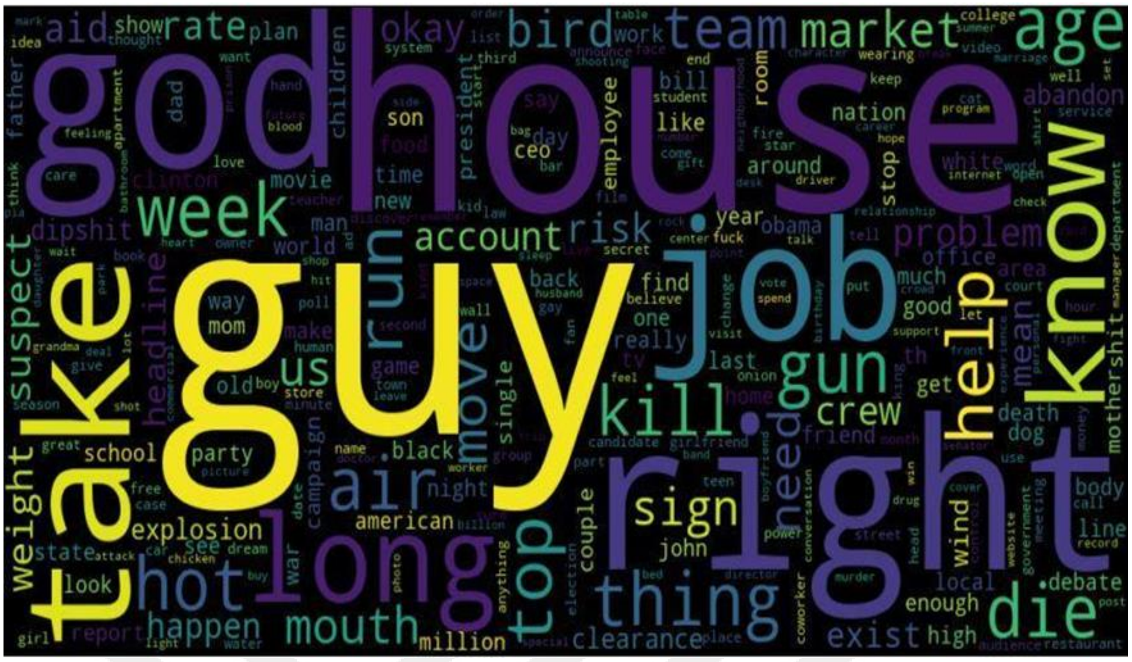


Figure 1.1: Tweet Cloud [77]

Mostly Sarcasm is often used to amuse, offend, make light of, or contradict anything. It is often associated with wit and irony. One employer said, "Kudos for your hard work, I have never been more impressed," for instance. Sarcasm in tweets is hard to identify when the individual isn't around. People use social media to interact with one another and to express their emotions since it is the era of social media. This platform is being used by a lot of organizations across the globe to better understand how people behave when it comes to products, movies, and political events [4]. People use sarcasm to communicate their opinions on social media; this makes it challenging for robots to understand human behavior as well as for the person expressing themselves who is not directly in front of them. To recognize sarcasm in a conversational situation, it may be important to not only comprehend the context of earlier remarks but also to have essential background information on the topic of conversation. Sarcasm detection has thus taken on significant importance in the current day. In order to identify whether a statement is ironic or not, sarcasm detection may be thought of as a binary classification issue [5]. Automatic sarcasm detection employs both machine learning and several feature techniques.

One of the biggest problems in the area of natural language processing is the recognition of sarcasm in text. Machines have a tough time identifying sarcasm since it is a subjective and context-dependent mode of communication. Lexical, pragmatic, and semantic elements have traditionally been used in sarcasm detection techniques, although these features have certain limitations when it comes to capturing the subtleties of sarcasm.

Deep learning techniques have attracted a lot of attention recently thanks to developments in natural language processing, particularly the ability to recognise sarcasm. Using deep learning techniques and contextual data, we provide a unique method for sarcasm detection in text in this study. For both contextual information and sarcastic signals to be recognised in the text, our method combines a bidirectional LSTM with an attention mechanism.

One key feature of our proposed model is the incorporation of user embedding, which captures the user's personality and stylometric characteristics. This is important because the manner of expression and the inherent variability of sarcasm can differ from person to person. The user embedding provides additional information to the model, enabling it to better understand the context and identify sarcasm.

The suggested model performs better when content-based features are extracted from embeddings using convolutional neural networks and Word2Vec. While performing NLP tasks, convolutional neural networks are frequently utilized to record contextual information and word associations. Word2Vec is a popular technique for representing words as high-dimensional vectors, which can be used as features in various NLP tasks.

The character limit for Twitter posts is typically 280, which is too few characters to give the best word co-occurrence statistics. For traditional learning algorithms that rely on factual highlights, such as term (TF-IDF) frequency-inverse document frequency or co-event, this informational gap poses a challenge. Deep learning designs have proven to be effective at managing common language because they can learn by modeling, provide flexible displaying options while minimizing component building, scale up to large amounts of data, use unlabeled data to detect drifts, and identify designs that are unreasonably complex for humans to understand [7].

The proposed models are evaluated by considering many parameters, and thus we found the best model, which performs very well on our dataset, and the model has been compared with all the six popular pre-trained models. As a result, the proposed Sarcasm detector model is very accurate, efficient, and potent. The detailed symbols and abbreviations used in this study.

1.2 SARCASM DETECTION ADVANTAGES

Sarcasm is frequently employed on social networking and microblogging websites, where users poke fun at or criticize in a way that makes it hard for even individuals to determine whether what is said or meant. Sarcasm is frequently cited as a barrier to sentiment analysis due to its metaphorical character [27]. Despite having a cheerful appearance, it conveys an inferred negative connotation. The study area of automated sarcasm detection has attracted interest due to the challenges of sarcasm and the benefits of sarcasm identification to sentiment analysis. Computer programs that assess whether a text is sarcastic are known as "automatic sarcasm detection" [25]. This encouraged other academics to incorporate sarcasm detection in other important fields. Sarcasm detection, for instance, can be used for a topic that deals with culture.

Joshi et al. [24] researched the factors affecting the accuracy of sarcastic predictions. The researcher claims that employing such a technique might aid in assessing the worth of brand-new data sets. Sarcasm detection was employed in a different piece of work by Kannangara [26] to classify people's political stances. The researchers presented three models in this respect for classifying the sociopolitical polarity of microblog entries. Together with other linguistic characteristics, the researchers also proposed a novel sarcasm recognition system that categorizes sarcastic views using ideology and fine-grained opinion as parameters. By employing the social media platform, sarcasm detection has a substantial industry deployment in addition to political use. These websites grow into substantial ecologies that enable users to freely express their opinions. Businesses use this ecosystem to give real-time customer support and to gather important public opinion about many aspects of their goods and services [29].

These companies also have a strong social media following and receptive marketing and customer service staff [28]. Using tools like HootSuite, businesses can manage a variety of difficult tasks like content management, sentiment analysis, and the extraction of pertinent messages for the company's customer service representatives to respond to, thanks to the enormous amount of data that social media platforms produce. These techniques, however, are not sophisticated enough to decipher complex verbal patterns like sarcasm that hide hidden meanings [28]. Sarcastic language can therefore be used to gauge a person's mood.

1.2.1 Sentiment Analysis

Determine the overall sentiment or emotion represented in a text or voice by using the technique of sentiment analysis [47]. Sentiment analysis seeks to pinpoint the attitude or emotion being expressed by the speaker or writer. The use of sarcasm may frequently make sentiment analysis more difficult since it can be employed to convey feelings that are at odds with the words' literal meanings.

Sarcasm identification can help sentiment analysis be more precise. By correctly identifying sarcasm in text, sentiment analysis algorithms can determine the true sentiment being expressed by the speaker or writer. This can help businesses and organizations to better understand the attitudes and opinions of their customers, employees, or stakeholders. For example, consider a customer review of a product that includes the following statement: "This is just what I needed, another thing to clutter up my house." Without sarcasm detection, sentiment analysis algorithms may interpret this statement as a positive review. However, with sarcasm detection, the algorithms can correctly identify the sarcastic tone of the statement and categorize it as negative. In another example, consider a social media post that reads: "Thanks a lot, now I have to spend my entire weekend cleaning up this mess." Without sarcasm detection, sentiment analysis algorithms may interpret this post as negative sentiment towards the person who caused the mess. However, with sarcasm detection, the algorithms can correctly identify the sarcastic tone of the statement and categorize it as a negative sentiment toward the mess itself.

Sarcasm detection can be particularly useful in social media monitoring, where sarcasm is often used to express opinions or comments on current events or trending topics. Social media monitoring involves tracking and analyzing social media conversations to gain insights into customer opinions, emerging trends, and other relevant information. By correctly identifying sarcastic statements in social media posts, sentiment analysis algorithms can provide a more accurate picture of the overall sentiment towards a particular topic or event.

1.2.2 Customer Service

Customer service is a crucial component of every organization, and being able to recognize sarcasm may help to improve the level of customer service. Sarcasm detection can help customer service representatives understand the tone and intention of customer interactions, leading to more effective and positive customer service experiences [34].

When a customer engages with a business, they may use sarcasm to express frustration or dissatisfaction. Without sarcasm detection, customer service representatives may misunderstand the true meaning of a customer's words, leading to ineffective or inappropriate responses. However, with sarcasm detection, representatives can identify the sarcasm and respond accordingly. For example, consider a customer who says, "Wow, thanks for taking forever to get back to me." Without sarcasm detection, a customer service representative may interpret this statement as genuine gratitude and respond positively. However, with sarcasm detection, the representative can understand that the customer is frustrated with the slow response time and respond with an apology and an explanation of what caused the delay. In another example, consider a customer who says, "Great job, now I have to wait another week for my package to arrive." Without sarcasm detection, a customer service representative may interpret this statement as a positive affirmation of their service. However, with sarcasm detection, the representative can understand that the customer is unhappy with the delay and respond with empathy and a solution to expedite the delivery.

Sarcasm detection can also help customer service representatives identify opportunities to provide additional support or assistance. For example, if a customer says, "I'm sure you have better things to do than deal with my issue," a representative with sarcasm detection can identify the sarcasm and respond by assuring the customer that their issue is important and will be resolved promptly.

1.2.3 Brand Reputation

Brand reputation is an essential aspect of any business, and the ability to detect sarcasm can play a crucial role in protecting and enhancing a company's reputation. Sarcasm detection can help businesses monitor and respond to online conversations, identifying negative sentiments and mitigating potential reputation risks [23].

Sarcasm is a popular way to communicate anger or unhappiness with a company, brand, or product on social media and other online platforms [56]. Without sarcasm detection, businesses may fail to identify and respond to negative sentiment, leading to reputational damage. However, with sarcasm detection, businesses can identify the underlying negative sentiment and respond proactively to address customer concerns. For example, consider a customer who tweets, "Great, my new product arrived broken. Thanks a lot, [company name]." Without sarcasm detection, a business may view this as a positive mention of its product and fail to address the customer's concerns. However, with sarcasm detection, the business can identify the underlying negative sentiment and respond with an apology and a solution to the customer's issue.

Sarcasm detection can also help businesses identify and respond to potential reputation risks before they escalate. By monitoring online conversations and identifying sarcastic statements, businesses can identify issues and concerns before they become major problems. This can help businesses proactively address customer concerns and mitigate potential reputational damage [45].

In addition, sarcasm detection can help businesses enhance their reputation by identifying and responding to positive sentiments [40]. By identifying positive sarcasm and responding with appreciation and gratitude, businesses can build stronger relationships with their customers and enhance their overall reputation for quality products and services.

1.2.4 Social Media Monitoring

Businesses may engage with their clients through social media, raise brand recognition, and boost sales [55]. Social media platforms may also be a source of unfavourable comments and criticism, which, if not handled properly, can hurt a company's reputation. Sarcasm detection can be a valuable advantage for businesses in social media monitoring, allowing them to identify and address negative sentiment in a timely and effective manner [23].

Figure 1.2: Social Media Monitoring [78]

Sarcasm is often used on social media as a way of expressing dissatisfaction with a product or service. Without sarcasm detection, businesses may fail to recognize the underlying negative sentiment and miss the opportunity to respond proactively to customer concerns. However, with sarcasm detection, businesses can identify negative sentiments and respond promptly, addressing customer concerns and mitigating potential reputation risks.

For example, consider a customer who tweets, "Wow, I just love it when my product arrives broken. Thanks, [company name]." Without sarcasm detection, a business may view this as a positive mention of its product and fail to address the customer's concerns. However, with sarcasm detection, the business can recognize the underlying negative sentiment and respond proactively to the customer's concerns.

Sarcasm detection can also help businesses identify trends and patterns in customer feedback, allowing them to improve their products and services. By analyzing customer feedback for sarcasm, businesses can identify common issues and concerns, helping them to make targeted improvements and enhancements.

1.3 SARCASM DETECTION IMPLICATIONS

Accurate sarcasm detection has significant implications for NLP and beyond. One of the key implications is the improvement of sentiment analysis. Sarcasm can significantly affect the sentiment of a text. For example, a positive statement that is expressed sarcastically can convey a negative sentiment. Therefore, accurate sarcasm detection can help improve the accuracy of sentiment analysis.

Another implication of accurate sarcasm detection is the improvement of social media monitoring. Social media platforms generate massive amounts of data every day. Accurately detecting sarcasm can help identify and monitor the sentiment and opinions of social media users on specific topics, which can be useful for social and political analysis.

Accurate sarcasm detection can also improve the accuracy of recommendation systems. For example, a recommendation system that fails to detect sarcasm may recommend products or services that are the opposite of what the user intends. Therefore, accurate sarcasm detection can improve the relevance and accuracy of recommendations.

Despite the importance of accurate sarcasm detection, there are still many challenges and limitations. One of the main challenges is the lack of standardized datasets for sarcasm detection. There is no widely accepted corpus or dataset for training and evaluating sarcasm detection models. Therefore, the development of accurate sarcasm detection models is hindered by the lack of data.

Another challenge is the cultural and linguistic differences in the expression of sarcasm. Sarcasm can be expressed in different ways across cultures and languages. Therefore, sarcasm detection models may not generalize well across different cultures and languages.

1.4 THESIS OBJECTIVES

The following are the main contributions of this research study:

i. Present a deep neural network-based contextual sarcasm detector that can recognize sarcasm in a given situation.

ii. Word2Vec for using embeddings and convolutional neural networks to extract the content-based features.

iii. The dataset of the headlines is collected from News in Brief (Sarcastic one) and the nonsarcastic headlines are collected from HuffPost.

iv. Various performance metrics, including precision, sensitivity, F1-score, and accuracy, have been used to compare the proposed model against earlier models.

v. The suggested models perform better than the alternative models across the board.

vi. A superb BERT classifier will be utilized to examine the impact of transfer learning on the effectiveness of the suggested model.

1.5 MOTIVATION

Research motivation for developing a deep learning-based sarcasm detector using context information is driven by the growing need for efficient and accurate methods for detecting sarcasm in online communication. Sarcasm is often used in online communication, particularly in social media, and it is essential to have tools that can accurately detect sarcasm to enhance the quality of online communication. The traditional methods of detecting sarcasm, such as lexical, semantic, and pragmatic features, have limitations in capturing the context in which the sarcasm is used. As a result, the accuracy of sarcasm detection using these methods is often low.

Moreover, the variability of sarcasm usage also poses a challenge for accurate detection. Sarcasm can vary greatly from person to person, based on their personality and stylometric characteristics. This makes it challenging to create a universal sarcasm detection system that can correctly identify sarcasm in all situations.

In order to get over these problems, we propose a contextual sarcasm detector based on deep neural networks that takes into consideration user embedding to capture the user's personality and stylometric attributes. The programme is trained on a substantial dataset of online interactions in order to correctly detect sarcasm from context. User embedding and context data are used to aid the model in understanding the context in which sarcasm is used, improving the model's accuracy.

Using a sizable dataset of online conversations, the suggested model has been tested and analyzed, and the findings reveal that it has a 91% accuracy rate. More study in this area is being driven by the model's effectiveness in identifying sarcasm from context, which will help to increase the precision of sarcasm detection in online conversation.

1.6 PROBLEM STATEMENT

Sarcastic remarks and opinions are frequent in today's environment when social media platforms are widely used. Sarcasm, on the other hand, is a subtly ironic style that is frequently misunderstood, resulting in misunderstandings and a communication breakdown. This is particularly true when reading written text because there are no non-verbal signals to help the reader grasp the words' intended meaning.

Sarcasm recognition has been a long-standing challenge in the study of natural language processing (NLP). The context is incredibly important in identifying the intended meaning of a remark, yet current algorithms for sarcasm detection only take lexical and semantic elements into account.

Given the increasing use of written text and the need to understand sarcasm, there is a pressing need for a more sophisticated and context-aware approach to sarcasm detection. This requires a model that can effectively capture the relationships between words and the context in which they are used, to accurately determine the intended meaning of a statement.

This research aims to address this challenge by developing a deep learning-based model for sarcasm detection from text with context information. The model will incorporate contextual information, such as the user's personality and stylometric characteristics, in addition to lexical and semantic features, to provide a more accurate and context-aware approach to sarcasm detection.

1.7 RESEARCH SIGNIFICANCE

The research on sarcasm detection using deep learning techniques and context information holds significant potential for improving the current state of NLP. By incorporating context information, the proposed model has the potential to overcome some of the challenges faced by previous models in detecting sarcasm, such as the lack of contextual understanding and the variability in sarcastic expressions. In a variety of real-world settings, including sentiment analysis, customer service, and social media monitoring, the suggested model may be used. Accurately identifying sarcasm can yield insightful information and facilitate better decision-making.

Moreover, this study might increase our knowledge of how context information affects sarcasm detection and benefit the area of natural language processing in general. The suggested model combines deep learning methods like LSTM and BERT, which can offer important insights into the potential of these methods for resolving challenging NLP issues. The research also highlights the significance of incorporating contextual information in NLP tasks and provides a framework for incorporating context information in other NLP applications.

Overall, the proposed model has the potential to significantly advance the field of sarcasm detection in NLP, providing a more sophisticated and accurate approach to detecting sarcasm. This study has the potential to advance our knowledge of how context information can be used to enhance the accuracy and efficacy of NLP models by illuminating the role of context information in the field of natural language processing and revealing insightful data about the potential of deep learning techniques.

1.8 THESIS STRUCTURE

The rest of the document is structured as follows: The related study of sarcasm detection is described in Section 2. The suggested models for sarcasm detection are described in Section 3. The experiments and their findings are presented in Section 4. Finally, the conclusion of the study with the future work is listed in Chapter 5.

2. LITERATURE REVIEW

The present state of the sarcasm detection field is examined in this section. The language ideas about sarcasm detection are examined first, after which the datasets used and the various methods are discussed. Furthermore covered is how various studies are performed. To reach conclusions and pinpoint any outstanding difficulties, the positives and negatives of each area are highlighted.

2.1 LINGUISTIC THEORIES OF SARCASM DETECTION

Linguistic approaches to sarcasm detection aim to identify linguistic cues that are characteristic of sarcasm. These cues can include words, phrases, and sentence structures that are typically used in sarcastic language. A speech act known as sarcasm is frequently utilized in ordinary conversation. Saying the reverse of what is intended in a fun or sarcastic manner defines it. Both human listeners and computational models have difficulty detecting sarcasm. Several linguistic theories have been proposed to explain the linguistic cues that signal sarcasm. These theories can be broadly classified into three categories: irony markers, modal verbs, and polarity cues.

2.1.1 Irony Markers

The irony is a complex linguistic phenomenon that is difficult to detect and understand. To detect sarcasm, researchers have looked at various linguistic markers that might indicate the presence of irony. One such marker is the use of irony markers. Irony markers are words or phrases that are specifically used to signal irony. These markers can include words such as "sarcasm" or "not really," or phrases such as "just kidding" or "as if."

The use of irony markers is thought to be one way to signal sarcasm. However, the use of irony markers is not always reliable, as they can sometimes be used for other purposes or not used at all. For example, someone might use the phrase "just kidding" in a completely serious manner, or they might not use any irony markers at all and rely solely on tone of voice and body language to signal sarcasm.

Despite these limitations, irony markers can still be a useful tool for sarcasm detection. By analyzing the presence and use of irony markers in text, researchers can get a better understanding of the use of sarcasm and how it functions in language.

2.1.2 Model Verbs

Another linguistic marker that has been studied in the context of sarcasm detection is the use of modal verbs. Modal verbs are verbs that indicate the speaker's attitude or degree of certainty about the action being described. Some common modal verbs include "can," "could," "may," "might," "should," "would," and "will."

Researchers have found that modal verbs can play a role in sarcasm detection. For example, sarcasm might be indicated by the use of modal verbs such as "should" or "could" in an ironic manner. For example, the sentence "You should try that new restaurant" might be sarcastic if the speaker thinks the restaurant is terrible.

2.1.3 Polarity Cues

Polarity cues are another important factor in sarcasm detection. Polarity refers to the positive or negative connotation of a word or phrase. Researchers have found that sarcasm often involves a reversal of polarity, where positive words are used in a negative manner or vice versa. For example, the sentence "That's just great" might be sarcastic if the speaker is expressing disappointment.

To detect sarcasm, researchers have looked at the presence of polarity cues in the text. Researchers can learn more about the tone and meaning of a statement by looking at how positive and negative words are used. They can use this information to judge if a statement is ironic or not.

2.1.4 Ambiguity

Ambiguity is another factor that can play a role in sarcasm detection. Ambiguity refers to the fact that a word or phrase can have multiple meanings or interpretations. In the context of sarcasm, ambiguity can make it difficult to determine the true intention behind a sentence.

For example, the sentence "That's just great" could be ambiguous, as it could be interpreted as either sincere or sarcastic. To detect sarcasm, researchers need to look beyond the words themselves and consider other contextual factors such as tone of voice, body language, and the relationship between the speaker and the listener.

2.2 AUTOMATIC SARCASM DETECTION APPROACHES

The automatic sarcasm detection problem is categorized on the basis of rule-based, deep-learning, and machine-learning methodologies.

2.2.1 Rule-Based System

Natural language processing and machine learning researchers are actively studying the automatic detection of sarcasm. Using a rule-based system is one method for sarcasm detection in written material. A system that employs a set of predetermined criteria to identify sarcasm in written text is known as a rule-based system. The categorization criteria used by rule-based systems [30] are obtained through hashtag analysis. The hashtags are tokenized, reduced to single words, and then matched against a Linux dictionary using criteria such as matching places, businesses, and currencies. They were able to reach an accuracy of 98% using this rule-based method.

How does a Rule-Based System Work?

A rule-based system for sarcasm detection works by analyzing the text for specific words or phrases that are commonly used in sarcasm. For example, a rule-based system might be programmed to recognize irony markers, such as words like "seriously," "literally," or "just kidding." When these words are detected in a text, the system can automatically identify the text as sarcastic.

Advantages

One advantage of a rule-based system for sarcasm detection is that it is straightforward to understand. Unlike more complex machine learning algorithms, a rule-based system can be easily programmed and maintained by a human expert. This makes it easier to understand how the system is making its decisions and to make changes to the system as needed.

A rule-based system also has the benefit of being relatively quick and simple to build. A rule-based system may be put into place relatively fast and with fewer data than machine learning algorithms, which need plenty of training data.

Limitations:

Despite its advantages, a rule-based system for sarcasm detection has several limitations. One limitation is that a rule-based system is limited by the rules that are programmed into it. If the rules are not comprehensive or are not updated regularly, the system may miss the sarcasm in the text.

Another limitation of a rule-based system is that it can be prone to false positive detections. For example, a rule-based system might identify a text as sarcastic when it is not, simply because the text contains one of the predefined irony markers.

2.2.2 Machine Learning-Based Systems

The use of machine learning for sarcasm detection has drawn a lot of interest in recent years. Much research has investigated various feature sets to train machine learning classifiers in this area [33]. The bag-of-words model, which depicts text as a collection of its words without taking into account their sequence or structure, is the most often used feature set for sardonic detection. Other research, however, has looked at many other feature sets. Using a huge corpus of 66,000 caustic tweets as an example, one research [32] employed pattern-based features. The patterns were created by spotting linguistic patterns that appeared often in the sarcastic tweets and utilizing those patterns as features to train an SVM classifier. Another study [33] made use of sentiment lexicon-based features, which rate the emotional potency of each word in the text with a polarity score. Next, together with other characteristics like emotions and user mentions, the sentiment lexicon-based features were utilized to train an SVM classifier. Another research [31] used a variety of indicators, including emotional elements, punctuation, parts of speech, word length, emotions, and semantic similarity. The features were derived using several emotional lexicons, and many classifiers, including Naive Bayes, Decision Trees, and SVM, were trained.

a. Naive Bayes Classifier

The Naive Bayes classifier, a probabilistic algorithm, employs the Bayes theorem to estimate the likelihood that a text is satirical. The algorithm is referred to as "Naive" because it believes that each attribute exists independently of the others. Despite this presumption, the Naive Bayes classifier has proven to be effective in real-world applications and is frequently employed in applications like sentiment analysis and text categorization.

The probability of a text falling into several classifications is determined by the algorithm using a frequency-based method. The likelihood of each word appearing in a sarcastic text is first calculated, and the total probability of the text being sardonic is then obtained by multiplying these probabilities. The anticipated class is then given the highest likelihood class.

b. Decision Trees Classifier

A tree-based method called the Decision Trees classifier employs a tree structure to describe a decision-making process. The interior nodes in the tree represent tests on features, whereas the leaf nodes in the tree represent class labels for each class. The process starts at the root node, moves up the tree, and stops at the leaf node after assessing the feature values along the way. The class label for that node is then updated with the anticipated class.

A strong method that can handle both continuous and categorical features, as well as intricate interactions between characteristics, is the Decision Trees classifier. It may be used for both binary and multi-class classification problems and can handle big datasets.

c. SVM Classifier

A linear method known as the Support Vector Machine (SVM) classifier separates the data into multiple groups along a line known as a hyperplane. The margin, or the distance between the hyperplane and the surrounding data points known as support vectors, is maximised by choosing a hyperplane that is near to those places. As the hyperplane's location is determined by the support vectors, they are essential to the classification process.

Whether performing binary or multi-class classification tasks, the SVM classifier can handle both linear and non-linear connections between the features. It has been widely used in a range of applications, including sentiment analysis, text classification, and picture classification, because to its high accuracy and robustness.

What is the Working of a Machine Learning-Based System?

In order to identify the patterns and characteristics of the sarcastic material, massive amounts of text data are examined using a machine learning-based sarcasm identification method. This data is used to train a machine learning system to make predictions about upcoming messages. For instance, a machine learning system may be taught using a big corpus of text that includes both sarcastic and neutral material.

Advantages:

A machine learning-based system for sarcasm detection has the benefit of being extremely accurate. Unlike rule-based systems, which are limited by the rules that are programmed into them, machine learning-based systems can learn patterns and make predictions based on a large amount of data. This makes them more effective at detecting sarcasm in text.

Another advantage of a machine learning-based system is that it can adapt and improve over time. As more data is fed into the system, the machine learning algorithm can continue to learn and improve its accuracy. This makes it possible for a machine learning-based system to continuously improve its performance and detect sarcasm in new and unpredictable ways.

Limitations:

A machine learning-based sarcasm detection system has a number of drawbacks despite its benefits. One drawback is that a lot of data is needed to train the algorithm. Particularly for individuals who are new to machine learning, this can be difficult and time-consuming.

A machine learning-based system's additional drawback is that it could be complicated and challenging to comprehend. Machine learning algorithms can be complicated and challenging to interpret, in contrast to rule-based systems, which are easy to comprehend. This makes it more challenging for a human expert to understand how the system is making its decisions and to make changes to the system as needed.

2.2.3 Deep Learning-Based systems

Researchers are looking at novel approaches to increase the precision of sarcasm detection as the discipline of natural language processing (NLP) develops [59]. Deep learning-based techniques have gained popularity as processing speeds have increased and the price of high-performance equipment has decreased. Particularly, sarcastic classifiers based on deep learning have been demonstrated to function rather well and are frequently described in the literature. Recent research, however, points to the possibility that a hybrid strategy that combines deep learning and machine learning might enhance the precision of sarcasm detection [60]. One method especially mentioned is to use BERT (Bidirectional Encoder

Representations from Transformers) to build BERT embeddings and include the semantics of the text [61]. Compared to conventional word embeddings, BERT is a pre-trained deep learning model that is better able to capture the context of the text. Many machine learning classifiers may be trained on the BERT embeddings once they have been acquired to increase the precision of sarcasm detection. Using a hybrid technique, which benefits from the best aspects of both deep learning and machine learning, it may be possible to identify sarcasm in text with even greater accuracy [62].

How does a Deep Learning-Based System Work?

A deep learning-based system for sarcasm detection works by analyzing large amounts of text data to learn the patterns and characteristics of the sarcastic text. This data is used to train a deep neural network, which can then be used to make predictions about new text. The neural network is designed to learn complex patterns in text data and make predictions about whether a given piece of text is sarcastic or not.

Advantages:

One advantage of a deep learning-based system for sarcasm detection is that it can be highly accurate. Unlike rule-based systems, which are limited by the rules that are programmed into them, deep learning-based systems can learn patterns and make predictions based on a large amount of data. This makes them more effective at detecting sarcasm in text.

Another advantage of a deep learning-based system is that it can handle complex and nuanced text. Deep learning algorithms are designed to handle complex patterns in data and make predictions based on those patterns. This makes them particularly well-suited for detecting sarcasm in text, which can often be subtle and nuanced.

Limitations:

Despite its advantages, a deep learning-based system for sarcasm detection has several limitations. One limitation is that it requires large amounts of data to train the algorithm. This can be challenging and time-consuming, especially for those who are new to deep learning.

2.3 RELATED WORK

2.3.1 Machine Learning Approach

Sarcasm and Irony, are used as synonyms. But the difference that irony is wider in its aspect and range, creates confusion in understanding. The research idea was the identification of potential features for categorizing irony and sarcastic phenomena in social media. The second research aspect was the differentiation of tweets from irony, sarcasm, or not from the above-mentioned behavior. The novel proposed model “emotIDM” in the paper is irony-aware [9].

Karthik Sundararajan and Anandhakumar Palanisamy have both proposed a "Multi-Rule Based Ensemble Feature Selection Model" to identify sarcasm in Twitter Data. The assignment was unique in that it divided sarcasm into four types. using a machine learning approach to assemble the ideal set of traits required for this classification. Also covered are several natural language difficulties. The dataset, which has a 92.7% accuracy rate [10], is made up of 76799 tweets.

Several feature sets are used by models to identify sarcasm. To create a rich-Support Vector Machine, the most prevalent and important characteristics are extracted. The main contribution of this work is the development of "multi-head attention-based Bidirectional Long-Short Term Memory" for corpus-based sarcasm detection. The use of two base layers, word-level encoding, and sentence-level multi-head attention is the key to this method. A task that is completed at the word-level layer is the summary of extra information from comments and directives. By keeping the various sentence components in mind, semantics may be understood at the sentence level. Results produced by this method perform better than the underlying model in every way [11].

By improving the feature engineering process, the performance of detection is improved in current models for sarcasm detection. Rule-based AI techniques were used as a traditional approach for detecting sarcasm in text, lexical and pragmatic features, stylistic features, situational disparity incongruity, or user-provided annotations. This paper addresses sarcasm detection with a multimodal approach and ensures that multi-modality is necessary for detecting sarcasm. The main thing was a novel data set consisting of 690 videos. Text, speech, and visual features were separately handled and performance was reduced to 12.9 % [12].

A similar work of manually labeled corpus is also used with annotated twitter dataset to compare results with human performance is also presented. Creating a corpus was a novel task, and annotations by accessing users' profile information and previous tweeted data. For experiments, both balanced and unbalanced datasets were used. The binary Classification technique was used with the logistic regression model [13].

2.3.2 Content Feature-Based Analyses

Content feature-based analyses refer to the methods of sarcasm detection that are based on the analysis of the features or elements within the content being evaluated. This can include analyzing the words used, the tone, the context, and any other relevant features in the content. The goal of this type of analysis is to identify patterns or characteristics that are commonly associated with sarcastic expressions and then use these patterns to automatically detect sarcasm. These methods can be implemented using various techniques, such as machine learning algorithms, natural language processing techniques, and text mining methods. The performance of these approaches can vary depending on the quality of the features selected and the algorithms used, but they are generally considered to be effective for sarcasm detection.

One of the first studies to use prosodic, spectral, and contextual cues to recognise sarcasm in speech was done by Tepperman et al. [44]. Their study piqued the curiosity of several sentiment analysis experts. The primary aim of the first textual studies was to identify lexical cues and syntactic signals that may be used as characteristics for sarcasm identification. Sarcasm detection was formerly considered to be a simple text classification problem.

Interjections, punctuation marks, intensifiers, and exaggeration, according to Kreuz and Caucci [54], are essential to the research. Moreover, Carvalho et al. [46] discovered that verbal and nonverbal emotions, such as emoticons and other keyboard symbols, are better indicators of sarcasm.

In Amazon product evaluations, Tsur et al. [45] discovered that exclamations like "yay!" or "awesome!" are good predictors, while Davidov et al. [47] used grammatical patterns like sarcastic hashtags to train classifiers using a semi-supervised method. Support Vector

Machines and logistic regression were used by Gonz'alez-Ib'anez et al. [48] to tweets, together with emoticons and snarky hashtags, as features. Also, they discovered that sarcasm was closely associated with both happy and negative emotions. Riloff et al. [49] refined this hypothesis by creating a bootstrapping method based on the idea that sarcasm is the difference between a good attitude and a poor circumstance (for example, "I enjoy being ignored"). Joshi et al. [50] applied a similar technique to predict implicit and explicit incongruity traits in a subsequent investigation.

Lukin and Walker [51] have proposed a bootstrapping method based on the extension of sarcastic N-gram signals like "no way," "oh really," and so forth, whereas prior investigations focused on unigrams. N-grams were also taken into account when Liebrecht et al. [52] and Ptáek et al. [53] created classifiers to analyse Dutch, Czech, and English tweets.

2.3.3 Context-Based Models

Context-based models refer to approaches for sarcasm detection that focus on the contextual information surrounding the content being evaluated. This can include information about the speaker, the audience, the situation, and any other relevant contextual factors. The goal of these models is to use this information to gain a better understanding of the intended meaning behind the words being used and to determine whether or not the expression is meant to be sarcastic [55].

Context-based models can be implemented using various techniques, such as machine learning algorithms, natural language processing techniques, and text mining methods. These models often perform well because sarcasm is often heavily reliant on the context in which it is used, and taking this into account can provide a more accurate understanding of the intended meaning. However, these models can also be limited in their ability to detect sarcasm in new or unseen contexts and may require extensive training data to perform effectively.

Contextual hints that were included in the tweets were previously recognized [35]. It has experimented with human interaction in two different ways. First, contextual hints are word patterns that are created when adjectives and adverbs are combined in a POS tag. Second, it's an essential pragmatic ability to be able to identify punctuation cues and interjections.

The human annotation of the NLP work in both cases was done using binary classes 0 or 1, which denote the presence or absence of characteristics. This study looked into the attributes that were manually retrieved after being annotated by humans. With non-literal expressions like "I slept well yesterday night," these characteristics acted as stronger signals of sarcasm, much like punctuation.

According to the author [35], there may be difficulties with terms that have a one-to-one mapping, such as the word order or sequence when translating into other languages. The author concentrated on the punctuation that was noted in several passages. As a result, it is crucial to recognize sarcasm using both pragmatic and POS elements, such as adjectives and adverbs.

Research by Lunando & Purwarianti [35] analyzed word feeling in phrases with overt sarcasm. To begin, the approach tokenized the words using SentiWordNet's unigram patterns. It is a lexical tool for words that rates words' sentiments on a scale of low, medium, and high. Next, using Google Translate, a translator converts the English language into Indonesian. Negative words were seen to be included in tweets where the following word was two or three words distant. When a word like "mahasiswa" (student) is followed by the word "harga," which is a neutral term, the word's meaning might change. This forms the phrase "harga mahasiswa," which changes the polarity from neutral to positive.

Yet sarcasm is more complex than just a lexical and syntactic phenomenon. Wallace et al. [36] have shown that a large number of the classifiers previously mentioned are ineffective when dealing with phrases that require context. For instance, with political postings, it is essential to have a broad understanding of the political climate of the country and region where the post was made to comprehend its significance.

Joshi et al. [20] used a more semantic method to predict implicit and explicit incongruity attributes using a mechanism similar to Riloff et al. (2013) [40]. In the context of a discussion forum post, they added the parent post, also referred to as the "elicitor," to the discussion thread. Rajadesingan et al. [56] and Bamman and Smith [57] collected contextual elements from the author's earlier tweets (user profile data). Moreover, Khattri et al[58] .'s search for contrasts between individuals and certain entities throughout the history of their tweets was an attempt to identify sarcasm.

2.3.4 Pattern-Based Methods

Sarcasm is a sophisticated approach for a speaker or writer to make an implied point [37]. The hashtag #sarcasm, which suggests that the tweet contains sarcasm, is used to gather search tweets. The training was conducted using Amazon review imbalance data, of which 471 reviews are positive and 5020 are negative, with a higher proportion of negative reviews in the sample. This ratio is predicted since, according to data based on the hashtag #sarcasm, there are more non-sardonic statements than sarcastic ones. After all, #sarcasm is a popular hashtag for online reviews [38].

With 4.12 ratings on average, the data more overtly show a propensity in this direction. The hashtag "sarcasm" was used in 1,500 tweets in the second data set [39]. The SASI (semi-supervised algorithm for sarcasm identification) method was created by the author. Moreover, it expanded the fields in which the idea from earlier studies might be used. To detect sarcasm, a small sample dataset of 80 favourable and 550 unfavourable Amazon reviews was employed. Since more negative than positive reviews include sarcastic cues, it chose unbalanced data. The content-based pattern characteristics, such as "[COMPANY] CW does not CW much" or "do not CW much about CW," were identified by the SASI algorithm. The high-frequency word [business] is represented by CW in this sentence. Because it happened regularly, like the "[COMPANY]" tag, it extracted the characteristics in predefined patterns.

To categorize sarcasm, the content-based patterns and punctuation characteristics from the data were put into the classification model KNN. Even when voice and punctuation features were included, the performance matrix F1 was 0.83 on Amazon reviews and 0.55 on Twitter tweets. Comparing the baseline findings to the combined feature strategy, which had a baseline + discrete features F1 of 0.83, the baseline results were not statistically significant. In order to identify sarcasm, combination traits performed best. The research was constrained by the small training dataset and the absence of meaningful results on the tweets. The tweets are collected using the #politics hashtag, and the SASI algorithm is then used to extract them. These tweets will input the sarcasm detection algorithm in the form of a vector. A big feature area is required to detect sarcasm using Part of Speech (POS) tags. The verb-verb and verb-noun features of a sentence are indicated by the POS tag. They are taken out in order to observe the polarity of the two-gram sentences. The polarity with positive or negative contrast with other verbs or phrases in a sentence is referred to as sarcasm.

González-Ibánez et al., et al. [40] The author used the Twitter API to gather sarcastic messages that were filtered based on the hashtag #sarcasm. The hashtags "happy," "joy," and "lucky" were used to gather positive sentiments, while "sadness," "anger," and "frustrated" to collect negative opinions. For each hashtag category, a total of 900 tweets were gathered. It retrieved lexical and pragmatic elements to recognize sarcasm. Lexical characteristics are specific positive words [41] that are selected from a lexicon with four categories and 64 entries for each category of positive phrases. These four categories include the spoken category, psychological process, personal concern, and linguistic process (adverb, pronouns).

Pragmatic features, such as emoticons and negative emotions, constitute a different category of the feature. Four distinct feature classes were proposed by the previous author: sarcastic positive (S-P), sarcastic positive-negative (S-P-N), sarcastic negative (S-N), and sarcastic non-sarcastic (S-NS). Whether a tweet was sarcastic or favourable was denoted by the letters S-P. Sarcastic tweets with either positive or negative polarities were denoted as S-N in a way similar to this, and sarcastic tweets with both positive and negative polarities were denoted as S-P-N. In this case, the positive tweets hinted at the tweet's positive polarity, but the negative tweets explicitly stated the tweet's polarity. Polarity is impacted by emotions. Positive emotion (Posemo), sarcastic negative emotion (Nemo), negative emotion (Negate), emoticons (Smiley, Frown), and auxiliary verbs (Auxvb) such is, are, and am are some of these emotions. Punctuation acts as pragmatic cues in a phrase, just like commas do.

Tweets with sarcasm typically contain more terms that express joy. Similar to this, the negative tweets contain more negative language (Negate is an important feature for S-P). To categorize these four categories and the unigram features, the author used the fundamental approaches of sequential minimum optimization, SVM, and logistic regression; however, bigram and trigram were not retrieved. A unigram is a single word or phrase, a bigram is two words or terms, and a trigram is three words or terms.

While training SVMs, the quadratic programming (QP) issue can be solved using sequential minimum optimization (SMO). The widely used LIBSVM programme, which is used to train

SVM, makes use of SMO. With F1 values of 0.71 for S-P, 0.57 for S-P-N, 0.65 for S-NS, and 0.69 for S-N, SMO beat other techniques for each kind of feature data. Five times in this inquiry did the validation criteria get utilised (Davidov, et al., 2010). A given data set is split into k parts or folds for K-fold cross-validation (CV), each of which acts as a test set.

Consider a 5-fold cross-validation case (K=5). Until every fold is included in the test set, this step is repeated. Sentences' words were tokenized using accessible dictionaries. The evaluation standards are built on dictionary-based lexica, including linguistic tags and vocal tokens created in tandem with word tokens from each tweet. Nevertheless, the problem is that because the integrated tool-based dictionaries only have a small number of terms, the majority of dictionaries do not cover linguistic patterns. Another problem is that these dictionaries are too tiny to encompass all word tokens; as a result, a larger dictionary covering all verbal patterns is needed.

A different researcher [42] concentrated on irony, comedy, and tweets relating to politics. The dataset consists of 50,000 tweets, 10,000 of which fall under each of the four categories of comedy, irony, technology, and humor. The words with the highest frequency (those that appear most frequently) were gathered from all papers using the following criteria: feature ambiguity, perplexity polarity, emotional situations (both imagined and pleasant), and sentence complexity. Using F1 values of 0.83 for humour vs. irony, 0.78 for humour vs. politics, 0.75 for humour vs. technology, and 0.72 for humour vs. general, the test was done using a Decision Tree (DT) with ambiguous characteristics. The performance was assessed using polarity, which was insufficient to detect humour. Instead, emotional elements were combined, such as irony, politics, technology, and comedy vs general since their respective F1 values were 0.62, 0.68, 0.61, and 0.56. The F1 scores for all features and combinations were 0.93, 0.86, 0.86, and 0.92. SVM and logistic regression are the other machine learning approaches, but no comparison with DT was made in the experiment; one may be made in the future.

The study [43] dealt with the categorization of sarcasm in both Czech and English. The properties of the 2-gram and 3-gram tweets were identified based on content, frequency, and pragmatic considerations. The 2-gram and 3-gram are two and three words, respectively, based on language patterns that appear repeatedly. It was demonstrated in the previous study that English tweets performed better at categorising than Czech tweets. As foundational elements, lexical tokens or words with patterns (such as punctuation) did well.

2.3.5 Automated Machine Learning (AutoML)

Automatic machine learning initially gathers characteristics from the variable-length input text and fixes them into fixed-length numeric vectors before performing its job (features). One further step is necessary before the vectors can be classified using machine learning. The bag-of-words representation is an illustration of this, in which each numerical feature reflects the count of a particular word drawn from the token's lexicon depending on its frequency. According to Islam et al. [64], reading about depressing people might cause a classifier to detect a person's pessimism. The feature extraction technique reflects the text and is crucial since it eliminates significant information loss. For instance, in the bag-of-words form, the word order is disregarded.

While performing well for classification, machine learning classifiers struggle for feature presentation. A mapping between word and term is necessary for dictionary-based approaches, which use dictionaries to represent features. Yet, easier to understand vector characteristics were produced using lexicon and rule-based strategies. The traits gleaned from these methods display the regulations. The baseline information is difficult for neural networks to convey when these data are transformed into vectors by embedding at deep levels. Nonetheless, it might be difficult to interpret the qualities.

The Auto-Sklearn framework was introduced by Feurer et al. [65]; it was the main study that concentrated on the probability model to determine the link between hyperparameters and model. The framework will analyse a hyperparameter like dropout during a model search iteration. The SMASH tree-based optimization approach was used to the model. SMASH is strongly advised as the quick cross-validation method. Early on, it eliminated underperforming hyperparameter values after evaluating one-fold. The AutoML framework was firstly made to operate better; a meta-learning phase was included to kick off the Bayesian optimization, increasing efficiency. A second addition made in this study was an automated ensemble creation stage that made use of every classifier discovered using Bayesian optimization. Eventually, our model used Auto-Sklearn to apply Bayesian hyperparameter to over 140 datasets of multiclass classification.

Using 10-fold cross-validation on two-thirds of the data, the SMAC optimization ran for 24 hours before saving the best performance, according to Hutter et al., [66]. As a result, it was determined that the configuration was appropriate for using the ML framework. The cost of computing to find the optimal values is the main issue.

Olson, et al. [67] provide a description of the tree optimization model's automated phase and data cleaning stage. Automation will start with the feature removal phase and build a new feature based on the present model. At the model selection stage, the training model is selected, and the parameters are then optimised to produce an accurate model. The model's validation phase trains the dataset. A tree-based pipeline optimization tool (TPOT), created through genetic analysis of the prostate cancer dataset, was used to carry out these automation steps. The automation technique TPOT was successful because it was able to spot cutting-edge pipeline operators, including synthetic feature constructions, which sped up the creation of competitive classifiers.

The Auto-WEKA framework, which was proposed by Kotthoff et al. [68] and is included in the Sklearn library, is a part of the Weka tool. Using the CASH optimization approach, the model was an additional attempt to identify the optimal model and its related hyperparameters. In comparison to other AutoML, this ML has numerous advantages. First off, Auto-Weka has regression methods to help with categorization. In addition, it facilitates WEKA's optimization of all performance measures. Lastly, it offers parallel runs (on a single computer) to identify practical configurations more quickly and preserve N best designs at each run rather than using best available configuration. Fourth, WEKA and Auto-WEKA 2.0 have been fully merged. Since Auto-straightforward WEKA's usage is its main selling feature. Thus, offering a push-button interface that doesn't need users to be familiar with the available learning model or the optimal hyperparameter It is also quick since just 1GB of memory was taken into account, in addition to user dataset independent memory. The Bayesian optimizer may explore more space throughout longer runs; production runs would take many hours. To accommodate impatient users, the AutoML framework's total running time was set by default to 15 minutes. The architecture was not stable enough to handle any deep learning models due to their lengthy runtimes. The second innovation was using Weka's grid search and multi-search packages to enhance hyperparameters. Nevertheless, you can only change one learner and one filtering strategy at a time using these programmes. Dropout is the sole hyperparameter that grid search improves, but it's a crucial one for the deep learning model. Whether or not dropout is required affects the model's performance. The advantage was that approaches could be used to improve performance of current deep learning models.

The Jin, et al. [69] proposal established the well-known AutoML-fastText framework, which was a departure from the prior AutoML framework's primary focus on deep learning tasks as opposed to shallow models. The cloud-based AutoML generated these settings using Docker containers, which were challenging to install and required more knowledge, and Kubernetes, an open-source technology for automating the deployment, scaling, and administration of containerized applications. A Docker container image is a small, independent executable file that contains the code and runtime necessary to execute an application. There is no need for a rich user experience because the API is simple to understand. Second, due to a lack of available computer resources, the cloud's graphics processing units (GPU) needed to be configured differently depending on the environment. The advantage of Keras is that it can be used by any CPU, as opposed to just cloud Dockers and Kubernetes. The Bayesian and Gaussian processes in the search method are used to optimise the model parameters. The module trainer is in charge of doing GPU training and trains the training data in many modules concurrently.

The suggested framework by Umemura et al. [70] that combines the many ML models creates two challenges. The first is how to include the various datasets and ML model interfaces; it is challenging to merge all models into a single, cohesive framework. Scheduling the model search among many computers is the second problem. This problem was resolved by the framework the author provided; it has four modules: driver, hyperparameters, tuner, scheduler, and a number of executors. An evaluation metric and a dataset are provided by the user. This will be sent to the hyperparameters tuner by the driver module. Hyperparameters were passed to the model search using the grid search. To choose the subset of configuration, the schedule must be performed after the driver has been activated to query it. The schedule will distribute each job according to profile data to evenly distribute the model's workload among executors. The driver module then applies Apache Spark training jobs to the executor. It generates trained prediction models and maps functions. The best model is chosen once these models have been educated by the executors' modules. The driver always begins and runs predictive models using a standard interface, and it completes all operations on the local machine to address the difficulties. The problem of data format is resolved by using uniformed format data, which contains vector numbers that will be translated into a predetermined format prior to training the model. The framework chooses the signals dataset SECOM and the physics event HIGGS dataset.

The framework will run model searches with 1 to 32 simultaneous tasks, timing each task's completion. For parallel runs, cluster machines are utilised. In multi-node clusters, Apache Spark is used to search the AutoML Sklearns model. The outcomes will be evaluated against other AutoML systems like spark-MLib and spark-Sklearn. Although this framework and the machine learning implementations share an interface, the dataset performed worse than AutoML. Due to the scheduling system's profile-based parallelism, the framework also has a degree of on-time completion of jobs. The degree of parallelism indicates how dependent a particular completion is on concurrent processes that operate concurrently with one another.

This study [71] used the AutoML to do semi-supervised learning (SSL). SSL, however, does not acquire the meta-features necessary for meta-learning, which use AutoML to accomplish semi-supervised learning. Additionally, SSL is a large separation technique that optimises the hyperparameters and improves performance. Because meta-features are largely irrelevant during the SSL process, the suggested method fills the gap between meta-features and SSL. It investigated the distribution of meta-features that have a direct impact on SSL. The SSL was trained more rapidly because to these meta-features, but fine-tuning is not supported. The hyperparameters big marginal technique was introduced in order to adjust the hyperparameters that choose the best model where the margin of the hyperparameters is higher. The model is at its best when these large margins are present. It has been established that SSL methodologies like SVM, TSVM, and other techniques are less successful than AutoML-SSL in terms of efficacy. The efficiency of Auto-Sklearn could not be compared to that of AutoML-SSL since the latter performed better on unlabeled data while the former only performs so with labelled data.

In order to test several text feature representation techniques for social media postings, Howard, et al [72] compared them downstream with Automatic Machine Learning

(AutoML) models. The reachout.com peer support forum provided the main dataset. Chen and co. [73] Postings that have been categorised for moderator attention in the context of self-harm or perceived suicide risk are included in the Self-reported Mental Health Diagnoses (SMHD) databases. They especially assessed how effectively the methods could classify posts that a moderator would designate as requiring an immediate response. It used 1,588 labelled posts that were gathered from the Reachout.com forum for the 2017 Computational Linguistics and Clinical Psychology CLPsych joint challenge. To represent posts, lexicon-based tools like Empath, Linguistic Inquiry, Valence Aware Dictionary and Sentiment Reasoner, Word Count, and others were used. DeepMoji, and Generative Pretrained Transformer-1 were three of the pre-trained artificial neural network models employed by the AutoML (GPT-1). It used the Tree-based Optimization Tool and Auto-Sklearn as AutoML tools to build classifiers for the articles. The GPT-1 model, which was enhanced with 150,000 unlabeled Reachout.com articles, served as the foundation of the most effective strategy. The top system achieved a new state-of-the-art performance with an averaged F1 score of 0.572 using the dataset for the CLPsych 2017 task. Nonetheless, the pretrain model result was not very significant. It was finished without using any extra metadata or previous posting data. According to error analysis, this top system usually overlooks indications of hopelessness. Moreover, it included pictures to aid readers in understanding the classifiers they had learned. This study discovered that transfer learning is a useful technique for assessing risk with only a small quantity of labelled data and that fine-tuning of pre-trained language models gave further advantages when significant volumes of unlabeled text were available.

It optimises the pre-processing pipeline and takes the pre-processing transformations into account in the prototype, claim Giovanelli et al. [74]. The SDSS dataset contains images of more than 25% of the known galaxies. Preprocessing pipeline prototypes use normalisation, a transformation technique that scales variables from 0 to 1. An alternative preprocessing technique called imputation substitutes an average or median value for missing data. In order to change the data, the preprocessing pipeline uses validation criteria instead of the baseline approach. It's also crucial to consider the transformation technique's sequence. Five transformation approaches were used in the framework with all potential arrangements of these prototypes. Also, following transformation, train the data using Bayesian, KNN, and random forest machine learning approaches. This approach's weakness is that it cannot handle the NLP preprocessing transformation used with deliberate training.

2.3.6 Deep Learning Approach

Hybrid neural networks are used as an additional method of sarcasm detection. News headlines data is employed, as opposed to the Twitter dataset, for sarcasm identification. as Twitter data is often unreliable. The news, however, has nearly no grammatical or spelling problems because it is created by professionals. The model is trained using 80% of the dataset. Hyperparameter tuning is a further phase in the procedure that raises the network model's performance by 5% in comparison to the basic model. Changes to the cost function's parameters improve efficiency. In addition, the inaccuracy is reduced by mapping input to output labels [14].

To identify between news that has been shared on Twitter and other sources, the article [15] emphasizes utilizing precise terminology. The purpose of this study is to modify the settings of existing models in order to use the most recent Transfer Learning models for identifying news clickbait. The author claims that this is the first attempt to alter transfer learning such that the clickbait community is excluded from media coverage. We have improved the BERT, XLNet, and Roberta models in this work by combining special parameters with adjustments to their automated structures, such as model extensions, pruning, and data augmentation techniques.

This article [16] suggests classifying Urdu text texts using the DSL Urdu News dataset, which has 662 examples from six different classes and is available to the general public. To test the effectiveness of transfer learning, the author used a BERT classifier, 10 machine learning, and 10 deep learning approaches. On datasets, the experiment was conducted. The ML classifiers Naive Bayes with NDM obtained the best performance of 94 percent on the DSL Urdu news dataset. For both datasets, the SVM classifier performed best, scoring 92% with NDM. For every dataset, the BERT classifier performed identically.

To provide a deep learning-based approach to identify irony in Hindi-English with a dataset of tweets, bilingual word embeddings from FastText and Word2Vec techniques were used. Among other deep learning models, CNNs, LSTMs, and Bidirectional LSTMs are used. The greatest results were obtained by attention-based Bi-directional LSTMs, which had an accuracy of 78.49% [17], outperforming state-of-the-art performances.

CNN and soft attention-based bidirectional long short-term memory are two further hybrid methods that employ punctuation for sarcasm detection. The goals of using this model were to make the sentiment-analysis process efficient and automate sarcasm detection by the best feature–engineered sets. Eight layers are used in the model, the first layer is for inserting tweets into the model. Second layer for mapping tweets to a low vector dimension. Third BLSTM layer, complex features are understood at this level by model. The attention layer, which is the fourth layer, is in charge of developing significant context. The fifth layer is where the convolutional job is carried out, and a new convolved feature set is added to the model (vector). Relu aids in modelling at this point in the process. The sixth layer is for the activation function. The max-pooling method is applied during downsampling. SoftMax activation is utilised in the last layer for output [18].

Sarcasm detection has made considerable progress in recent years. Text classification is often seen as a standard issue for this use. Convolutional Neural Network is used for this purpose. The key difference between other models is the removal of discrete or complex feature engineering. Disadvantages of complex feature sets include the requirement of professionals for feature design. By avoiding the manual feature set, the neural layer automatically inserts features and secondly, real-valued word vectors data is used for improving efficiency. A dataset of 8727 tweets, consisting of conversation, basic, and history tags are used [19].

For words and characters sentiment lexicons, hidden patterns, and markers of text are used for text classifiers with various features. The major drawbacks of these techniques are the process of manual feature generation to improve performance. The method discussed in the paper is developed by Deep Neural Network. Without the usage of manual feature set extraction, it works on lexical representations via the convolutional layer. Then it learns and leads on embedded user’s data. The main thing required in this model is some details from previous posts [20].

A bi-directional Long Short-Term Memory for word-level information, a layer for embedding, and CNN are all used in an architecture designed for sarcasm detection. The first layer of the neural network is fed with news headlines. The next step is to encode each word in the input, converting it to a vector. The utilized dataset consists of 26,709 entries, out of which 56.1% are authentic and the rest are fake. The accuracy rating might be reached at 86.16%. The lack of data availability is the primary concern raised. Large datasets are where deep neural networks perform best [21].

In a study conducted by the author, deep neural networks (DNN), recurrence neural networks (RNN), and convolution neural networks (CNN) were employed for multiclass classification [22]. A dataset with 0.15 million tagged Urdu words was used in the study. It was recommended to use word embedding, inverse document frequency, term-frequency, and one-hot encoding techniques (TF-IDF).

Results of the investigation showed that the DNN classifier had a promising accuracy of 84%. This shows that multiclass categorization using deep learning models can be done successfully in the Urdu language.

Table 1 provides a detailed literature summary of this study, including the models and techniques used, the dataset size and origin, the feature selection methods, and the resulting accuracy. This work contributes to the body of knowledge already available on multiclass classification and deep learning, specifically in the context of the Urdu language.

Sentiment analysis, text classification, natural language processing, and other areas and applications can use the findings of this study. More precise and effective multiclass classification in the Urdu language may be achieved by applying deep learning models and combining different feature selection techniques.

Table 2.1: Summarization of Literature Review.

Paper	Dataset	Methodology	Accuracy
Bilingual word embedding is used to detect sarcasm in Hindi-English code-mixed data	Hindi-English tweets dataset	Bilingual word embedding derived from FastText, Word2Vec CNNs, LSTMs, and BI-LSTM	Accuracy 78.49%
Model for Sarcasm Feature Selection Based on Multiple Rules	76,799 tweets dataset	Model for Ensemble Feature Selection Based on Multiple Rules	95.98%, 96.20%, 99.79%, and 86.61%, respectively, for courtesy, rudeness, rage, and deadpan sarcasm
Twitter irony and sarcasm detection	Tweets Dataset	Novel proposed model "emotIDM	Accuracy 86.61%
Using a mixed neural network to detect sarcasm	News dataset which contains news headlines	Hybrid Neural Networks (CNN, LSTM)	Accuracy 87 %
Sarcasm Feature Selection Model using Multi-Rule Based Ensemble.	Twitter Dataset with Sarcasm Hashtag	Multi-head attention based BI-LSTM Model and sentence level multi-head attention	Good Accuracy with 89 %
Sentiment Analysis Using Sarcasm Detection in Indonesian Tweets	News Headline 26,000	CNN, a layer for embedding, and BI-LSTM	Accuracy 86.16 %

Table 2.1: Summarization of Literature Review" Tables Continued."

Automatic irony- and sarcasm detection in Social media	Amazon and Twitter dataset	Natural Language Processing (NLP) and Machine learning models that classifies text as sarcastic or non-sarcastic.	Accuracy 87% and 71% for the Amazon products and the tweets reviews respectively
A Machine Learning Approach to Text-Based Sarcasm Detection	Kaggle dataset	Dummy Classifier, Logistic Regression, and SVM	80% Accuracy
A deep learning approach for identifying sarcasm in text	Twitter Dataset	A rule-based classification method	very high accuracy
A novel Auto-ML Framework for Sarcasm Detection	Irony Dataset	SVM KNN CNN	0.90 accuracy for SVM 0.88 accuracy for KNN 0.78 accuracy for CNN

3. METHODOLOGY

This chapter discusses how the study was carried out using research methods, designs, environments, instruments, and techniques. This chapter discusses the approach and methodology used to establish this research project.

3.1 ADOPTED METHODOLOGIES FOR SARCASM DETECTION

In this manifold research, as per figure 3.1. A news headlines dataset is used that is downloaded from Kaggle. Firstly dataset is preprocessed and then given to Deep Learning models LSTM and CNN to detect the sarcasm after that the valuable feature will be extracted and passed to the BERT classifier.

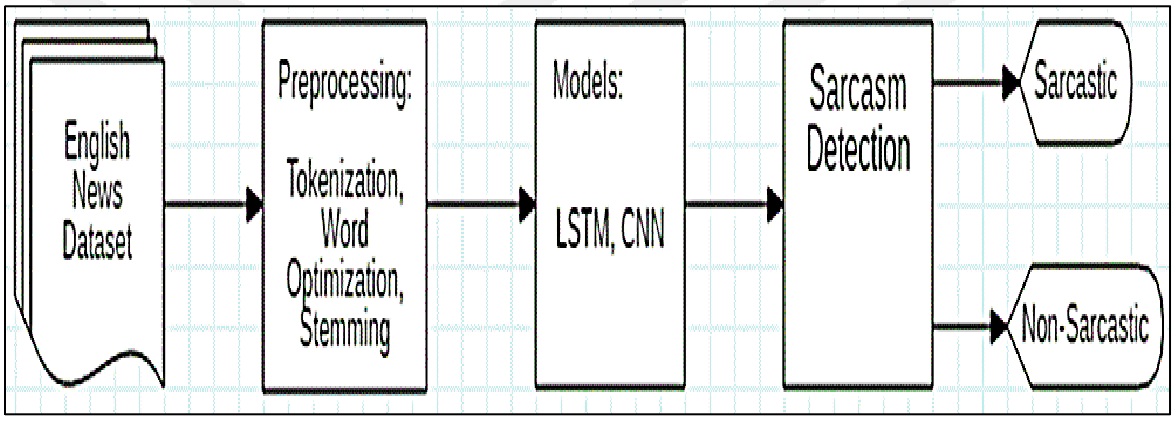


Figure 3.1: Adopted Methodology Architecture [20].

3.1.1 Dataset Description

To reduce the noise in the data, we selected the dataset from Kaggle having news headlines in the data. As we know, the news is written in clear English with proper headlines. The dataset comprising the headlines is collected from News in Brief and all the collected news is sarcastic. Other non-sarcastic headlines are taken from HuffPost. The general details of the dataset are given in Table 2. We can say that the text in the headlines has much more formal language.

Table 3.1: Dataset Description.

Dataset Description	
Instances	28,620
No of Sarcastic Records	13,634
No of Non-Sarcastic Records	14,986

3.1.2 Dataset Preprocessing

The purpose of data preprocessing is to improve the data or make it more compliant with the model. Several data preparation methods may be used to improve data. We cleaned the data for this study project first, and then we looked at the most frequent terms. The Dataset also no longer contains the stop words. We tokenized the data following the removal of the stop words. Data cleansing is a fundamental process that CNN systems frequently use to guarantee numerical stability. Normalized CNN models are more likely to train more quickly and descend with a stable gradient. After text cleaning, we only have 205988 words left for this study.

3.1.3 Proposed Model Architecture

The original model [20] used pre-trained user embeddings (for context) and tweets (for content) as input and provided a binary (Sarcastic or not) output. We modified our model to remove the user embeddings because it is claimed in this dataset that sarcasm is not dependent on authors but rather on common knowledge and current events. At each level, a new model is used to encode the word's context in phrases (LSTM). A new Attention module has been added to the LSTM module, adding additional weights to the encoded context at each step.

In the original architecture, the CNN module most likely included the encoded context from the LSTM module. Figure 1 displays the detailed model. The bi-directional LSTM attention module is comparable to the alignment and translation modules used in the Neural Machine Translation Task. The direction of a bi-directional LSTM's forward and backward motion. The backward LSTM calculates the hidden states in a forward fashion while the reading order is reversed. The order of progressing hidden states is determined by the forward LSTM.

The forward and backward states of each word from the input sentences are combined to create the annotation. The annotation hj thus includes summaries of the words both before and after it. The comment anytime step will focus on the words in the information phrase that come right before it since LSTMs tend to talk to ongoing input more readily. With a focus on the segments that include the comparing input word, each concealed state provides information about the whole input sequence. The weighted total of each of these annotations is the context vector c.

$$c = \sum_{i=1}^{N} \propto_i h_i \tag{4.1}$$

Using the ratings of each hidden state, Softmax calculated the weight/attention for a hidden state h I in this scenario. By putting each h I through a multilayer perceptron, which produces a score, a score is generated for each h I. The context vector c is now present in the CNN module's output. Typically, this vector of critical properties is then sent into an MLP, which yields a binary probability distribution of the sarcasm of the statement.

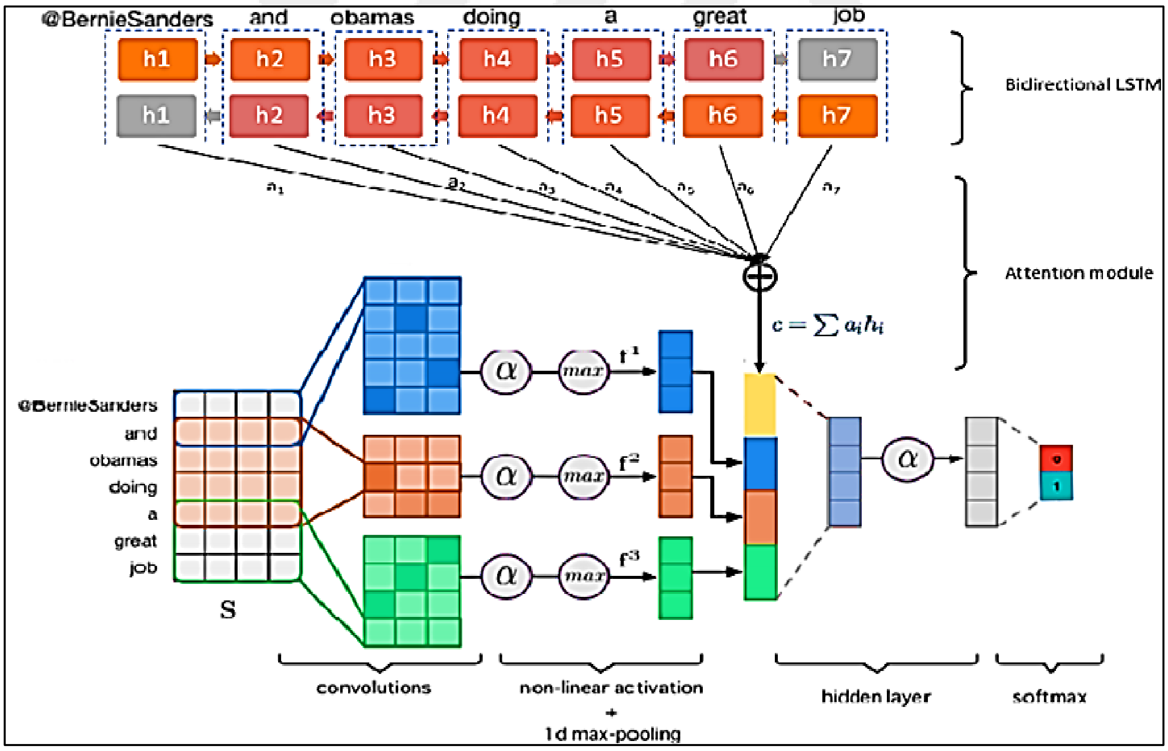


Figure 3.2: Softmax View [20].

Bidirectional Encoder Representations from Transformers (BERT) are evaluated using base vocabulary and vocabulary produced using a top filter-based feature selection approach since there is a dearth of research on how to improve BERT's performance across target tasks.

Also included are crucial recommendations for enhancing BERT's text categorization performance. This makes it simpler to evaluate which parameters and their corresponding values are most significant for enhancing sarcasm detection [23], as well as how successful a language model's vocabulary constructed using a feature selection approach based on top filters is.

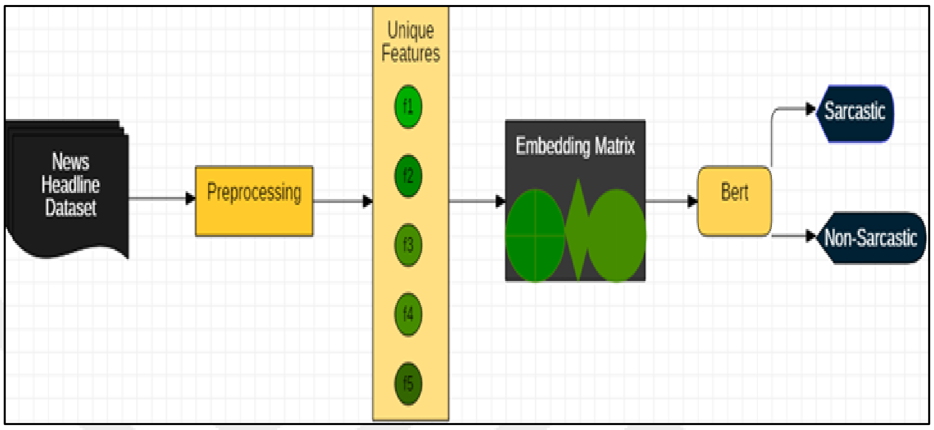


Figure 3.3: BERT Model Architecture [63]

The effectiveness of transfer learning for on-the-job sarcasm detection using the pre-trained language model "BERT" is examined in this part. Utilizing pre-training language models has proven to be quite beneficial for learning broad language representations. This may be achieved by language models like BERT by producing several embeddings for a single word that appear in various contexts. both from left to right and from right to left to anticipate the word that comes after it, but not both at once. BERT, in comparison, uses the entire phrase to learn from all of the words that are scattered throughout, as seen in figure 3. Before providing a forecast, it masks words in a certain context at random. It is substantially more exact since it uses transformers. Although BERT has shown promise in several NLP tasks, insufficient research has been done to optimize it for the advancement of the intended NLP tasks. This paper covers the application of BERT for text document categorization in great detail. To enhance BERT's effectiveness at sarcasm detection, we investigate several tuning techniques.

4. EXPERIMENTS AND RESULTS

When working with machine learning models, it is often important to establish a baseline level of performance before making any changes or updates. In the context of sarcasm detection, this might involve starting with a clean dataset of English text that can be used to train a new model. Once this baseline has been established, it may be possible to make adjustments or improvements to the model based on the results.

In the case of the model being discussed, the authors noted that they had updated it with a new dataset of clean English text. As a result, this new dataset was considered the baseline for the model. In addition, the authors removed author embeddings from the model because the new dataset suggested that sarcasm was not dependent on individual authors. Instead, the model was based on common knowledge and ongoing events that could be used to detect sarcasm in any text.

By removing author embeddings from the model, the authors were able to simplify the model and focus on the key features that were most important for sarcasm detection. This allowed them to develop a more streamlined and effective model that could be used in a wide range of contexts. By starting with a baseline and making targeted improvements to the model over time, it may be possible to develop more accurate and reliable models for sarcasm detection in the future.

4.1 EXPERIMENTAL SETUP

In this research, the authors describe the use of pre-trained word2vec embeddings for word representation. In the absence of missing data, words are initialized equally and randomly in the model during training. The dataset was divided into training, validation, and testing groups at a ratio of 80:10:10. The authors then adjusted various hyper-parameters, including output channels, hidden units, filter width, learning rate, dropout fraction, grid search, and model regularization.

During training, the model was optimized by lowering the cross entropy between the real table and the model's predictions. The backpropagation algorithm was used to calculate the gradients of the network parameters, which were then updated using the Adagrad rule. This process allowed the model to learn and improve over time, ultimately resulting in a more accurate and reliable tool for detecting sarcasm in text.

By using pre-trained word2vec embeddings, the authors were able to take advantage of the wealth of knowledge and expertise that had already been developed in the field of natural language processing. This approach allowed them to avoid the time-consuming and resource-intensive task of training their word embeddings from scratch. Instead, they could focus on fine-tuning the model and optimizing its hyper-parameters to achieve the best possible performance.

Throughout the process, the authors paid close attention to the various hyper-parameters that were adjusted. This included output channels, which control the number of output filters in the convolutional layer, and hidden units, which control the number of hidden units in the fully connected layer. The authors also adjusted the filter width, which controls the number of words that are considered at a time during the convolutional layer. In addition, they optimized the learning rate, dropout fraction, and regularization to further improve the model's accuracy and generalizability.

Overall, the authors used a rigorous and systematic approach to developing and fine-tuning their sarcasm detection model. By carefully selecting and optimizing various hyper-parameters, and by using pre-trained word2vec embeddings to represent words, they were able to achieve a high level of accuracy and reliability in their model. These findings may have important implications for future research in natural language processing and sarcasm detection, as well as for practical applications in fields such as social media analysis and sentiment analysis.

4.2 MODEL RESULTS

The authors of this study evaluated the performance of their proposed method for sarcasm detection and compared it to a baseline method. They used a balanced dataset, which means that the number of sarcastic and non-sarcastic comments was approximately equal, to ensure that the model was not biased towards one particular class.

To evaluate the accuracy of the proposed method and the baseline, the authors conducted hyper-parameter tuning to optimize various settings such as output channels, hidden units, filter width, learning rate, dropout fraction, grid search, and model regularization. The model had to be adjusted using training data, and its performance had to be tested using validation and testing sets.

The final accuracy results were published by the authors in Table 4.1 after hyper-parameter tweaking was finished. The accuracy of the proposed technique and the baseline were broken down in this table in terms of false positives, true positives, true negatives, and false negatives.

The authors were able to assess the efficacy of their strategy by contrasting the accuracy results obtained using the suggested method with the baseline. They discovered that their suggested strategy performed better than the baseline in terms of accuracy, showing that it was better at spotting irony in the text.

Table 4.1: Results of Model

Implementation	Accuracy
Baseline	89.5%
CNN with LSTM	91.0 %
Proposed Method BERT	72 %

The use of pre-trained language models has become increasingly popular in the field of natural language processing. One such model is BERT, which has been shown to excel at using a wide vocabulary to generate accurate and meaningful representations of text.

The Word Piece model is the foundation of the BERT tokenizer, and it creates a fixed-sized vocabulary by greedily building the most prevalent words, sub-words, and individual characters that best match a given language's data. By creating embeddings of sub-word tokens that maintain the bulk of the contextual meaning of the words and individual characters, BERT can successfully manage sentences that are not part of the vocabulary.

BERT produces useful representations for the features contained in experimental datasets by leveraging its ability to handle out-of-vocabulary (OOV) words. This is particularly useful in cases where the dataset includes rare or uncommon words that are not part of the standard vocabulary.

In a recent study, BERT was used to analyze a headlines dataset, and the results were promising. The dataset demonstrated a 72% positive return when using BERT, indicating that the model was effective at detecting and understanding the nuances of the text.

The use of BERT and other pre-trained language models represents a significant step forward in the field of natural language processing. These models offer a powerful tool for analyzing large datasets and generating accurate and meaningful representations of text. By leveraging the latest advancements in machine learning and artificial intelligence, researchers and practitioners can gain new insights into the complexities of human language, and ultimately improve our ability to communicate and understand one another.

4.3 CONFUSION MATRIX PARAMETERS

In the field of machine learning, performance evaluation is crucial in determining the effectiveness of a model. One popular method for evaluating the performance of a classification model is the use of a confusion matrix. A confusion matrix provides a clear and concise way to visualize the accuracy and errors of the model's predictions by comparing the predicted class labels against the actual class labels.

The confusion matrix lists the number of labels for all classes that are actual and predicted. The classes are typically defined based on the problem being solved, for example, in a binary classification problem, the two classes could be positive and negative.

The four possible outcomes of a binary classification problem are true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP represents the number of correctly classified positive instances, TN represents the number of correctly classified negative instances, FP represents the number of negative instances that were incorrectly classified as positive, and FN represents the number of positive instances that were incorrectly classified as negative.

Fig. 3 shows the confusion matrix generated for the model in this study. The confusion matrix provides a visual representation of the model's performance, allowing us to quickly and easily assess the model's strengths and weaknesses. For instance, we can see from the matrix that the model achieved a high number of true positive and true negative classifications, which indicates that the model was able to accurately classify most of the instances. However, the model also had a moderate number of false positive and false negative classifications, indicating that there is still room for improvement in the model's accuracy.

By analyzing the performance metrics values generated from the confusion matrix, researchers and practitioners can gain valuable insights into the effectiveness of the model and identify areas for improvement. The use of a confusion matrix is a simple yet powerful tool for evaluating the performance of classification models, and its widespread use demonstrates its effectiveness in the field of machine learning.

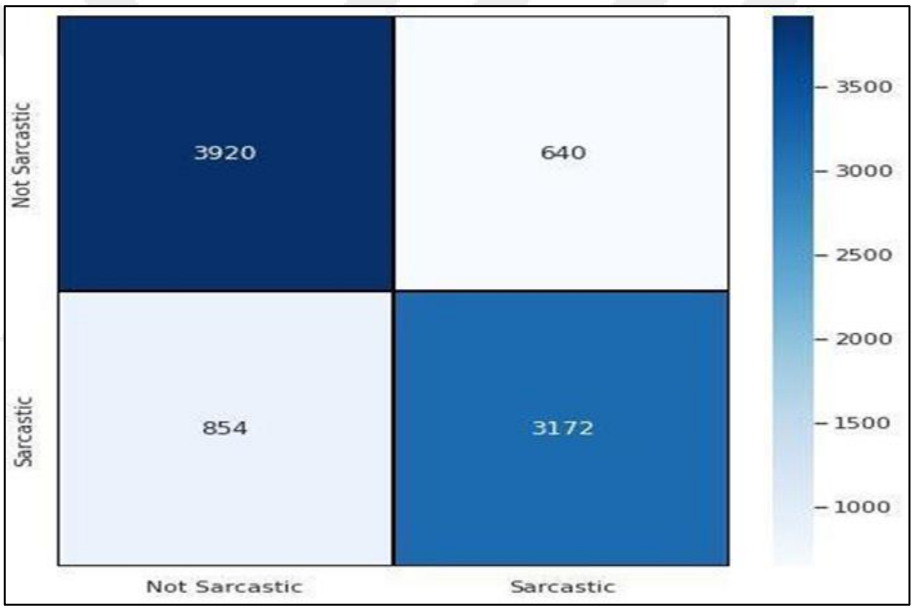


Figure 4.1: Confusion Matrix [22]

4.4 ACCURACY

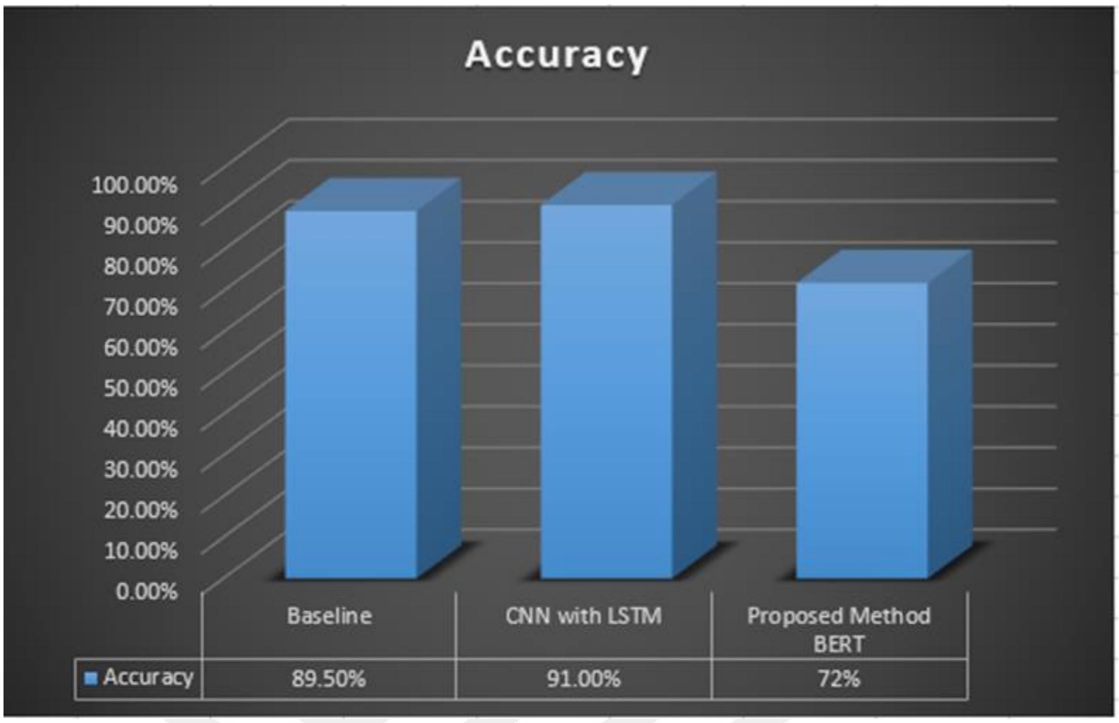
In the field of machine learning, accuracy is one of the primary evaluation metrics to measure the performance of a model. It is the ratio of the number of correct predictions (true positives and true negatives) to the total number of predictions made. In this study, accuracy is used to evaluate the performance of the proposed model for detecting sarcasm.

The baseline model had an accuracy of 89.5%, which means that it correctly identified 89.5% of the sarcastic and non-sarcastic comments. However, since the baseline model lacked contextual features, it could not perform as well as the proposed model. In the proposed model, LSTM was used to extract features, which were then passed through the CNN model and BERT classifier. This model outperformed the baseline model, achieving an accuracy of 91%.

The increase in accuracy can be attributed to the proposed model's ability to extract contextual features and use them to make more accurate predictions. By incorporating LSTM, the model can understand the context of the sentence, which is crucial for identifying sarcasm. The CNN model then extracts content-based features that are used by the BERT classifier to make the final prediction.

The use of the BERT classifier also contributed to the improved accuracy of the proposed model. The BERT classifier is a pre-trained language model that can handle complex language structures and outperforms other models in various natural language processing tasks. In this study, the BERT classifier achieved an accuracy of 72%, which is lower than the proposed model's accuracy. However, it still demonstrates the effectiveness of the BERT classifier in detecting sarcasm.

Overall, the suggested model outperformed the baseline model in terms of accuracy, highlighting the value of adding contextual data and utilizing a strong classifier like BERT to identify sarcasm in text. The study's findings can be applied to enhance the precision of other natural language processing models and software.

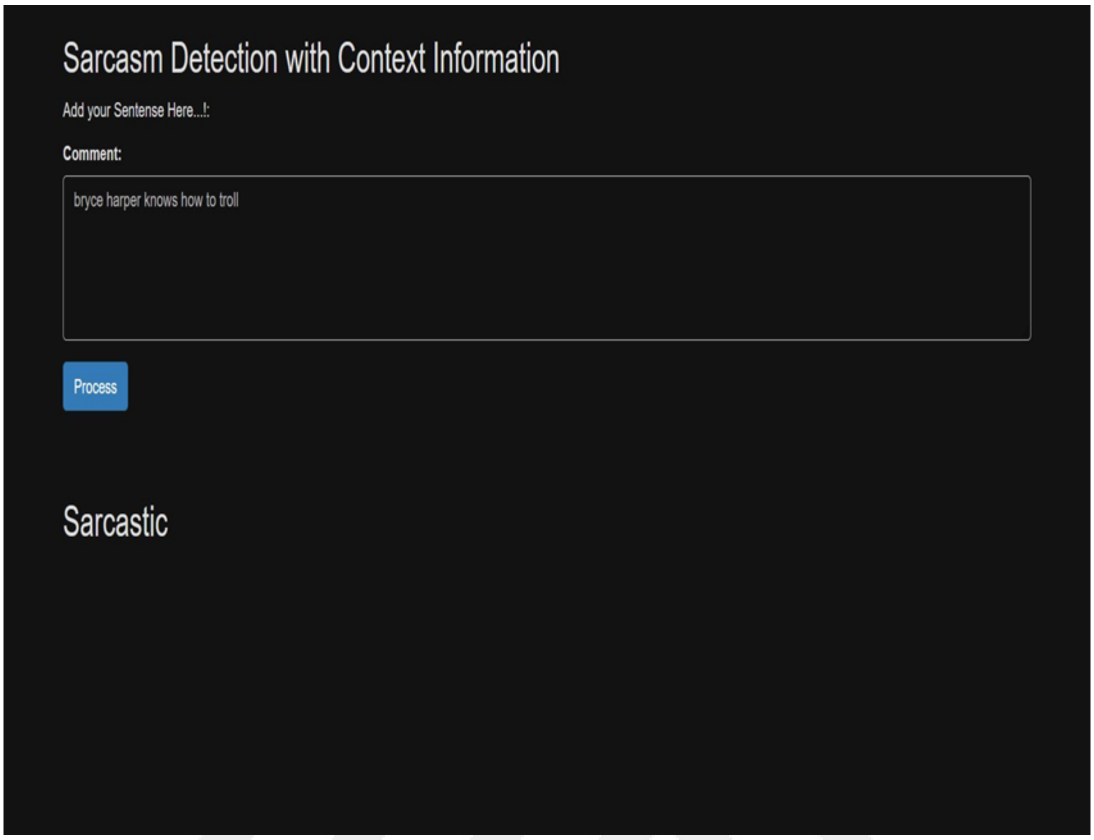


	Baseline	CNN with LSTM	Proposed Method BERT
■ Accuracy	89.50%	91.00%	72%

Figure 4.2: Accuracy Graph

4.5 GUI

In the following figure, users can enter text in the large text box and click the "Process Text" button to detect sarcasm. If the system detects sarcasm in the input text, the GUI screen displays a message indicating that sarcasm has been detected and provides context information about the specific statement that was detected as sarcastic. In this example, the context is "I just love getting up early on weekends." Note that the specific implementation and appearance of the GUI may vary depending on the specific application and design choices.


Sarcasm Detection with Context Information

Add your Sentense Here...!:

Comment:

bryce harper knows how to troll

Sarcastic

Figure 4.3: GUI Screen of Our Proposed System.

5. DISCUSSION AND CONCLUSION

5.1 CONCLUSION

Researchers and psychologists have been fascinated by sarcasm for a long time since it is a sophisticated type of communication. Sarcasm is a typical human behavior that is frequently utilized to communicate one's unfavorable viewpoint positively. It is a hilarious or ironic style of mocking or criticizing something or someone. It can also be used to parody or playfully show rage or irritation. It is a type of verbal irony that is employed to imply the exact opposite of what is intended. To identify the true intentions behind sarcasm, many previous models have been developed to detect sarcasm based on syntactical, lexical, and semantic analysis of the user-generated text. These models have been successful in detecting sarcasm in many cases, but they have limitations. For example, they do not take into account the context in which the sarcasm is being used, which can lead to inaccurate results.

In this paper, the authors proposed a new approach to detecting sarcasm that takes into account the contextual information in which it is being used. They developed a contextual sarcasm detector that uses deep neural network techniques to identify sarcasm with contextual information. The proposed solution utilizes user embedding that encodes the personality and stylometric features of the user. The authors used Word2Vec to generate embeddings and then used convolutional neural networks to extract the content-based features.

The authors found that their proposed solution gave an accuracy of 91%, which is more accurate than the base model. The authors also evaluated the transfer effect using a BERT classifier, which gave an accuracy of 72%. This suggests that the suggested method outperforms the most recent state-of-the-art models in sarcasm detection. The writers found that sarcasm and how it is expressed differently from person to person. As a result, the suggested method makes advantage of user embedding to encode the user's personality and stylometric characteristics. This method enables the sarcasm detector to be more accurate by allowing the system to understand the distinctive qualities of each user.

The authors also noted that their proposed solution has several limitations. For example, it relies on user-generated text and may not work as effectively for other forms of media, such as images or videos. Additionally, the system may not be effective for detecting sarcasm in languages other than English.

In conclusion, the proposed solution is an innovative approach to detecting sarcasm that takes into account the contextual information in which it is being used. The sarcasm detector's accuracy is increased by using user embedding and deep neural network approaches. While the proposed solution has some limitations, it has the potential to be a valuable tool for researchers, psychologists, and others interested in understanding the complex nature of sarcasm.

5.2 DISCUSSION

Sarcasm is a powerful tool in communication that has the ability to convey a message that is opposite to its literal meaning. Sarcasm detection is a challenging problem in natural language processing due to the contextual nuances and variations in the use of sarcasm in different domains. With the growing volume of text data on the internet, sarcasm detection has become an essential task in various applications such as sentiment analysis, opinion mining, and social media monitoring. In this article, we will discuss the use of deep learning techniques for sarcasm detection from text with context information.

Sarcasm detection is a challenging task in natural language processing due to the contextual nuances and variations in the use of sarcasm in different domains. Sarcasm can be expressed in a variety of ways, and it may depend on the context and the speaker's intention. Sarcasm can also be ambiguous, as it can be used for different purposes, such as to convey humor or criticism. Sarcasm can also be mixed with other forms of speech, such as irony or metaphor.

Traditional approaches to sarcasm detection rely on linguistic features such as negation, hyperbole, and rhetorical questions. These features are used to identify the presence of sarcasm in text. However, these approaches have limitations, as they rely on hand-crafted features that may not generalize well to different domains and languages.

Deep learning techniques have been shown to be effective in sarcasm detection from text with context information. Deep learning models can learn to automatically extract relevant features from text data and can handle the variability and complexity of sarcasm in different domains. Deep learning models such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have been used for sarcasm detection and have shown promising results.

Contextual information can provide additional cues for sarcasm detection. Contextual information can include information about the speaker, the audience, the topic, and the discourse structure. Contextual information can be incorporated into deep learning models to improve sarcasm detection performance. One approach is to use attention mechanisms that can focus on relevant parts of the input text based on the context.

Sarcasm detection has various applications in natural language processing, including sentiment analysis, opinion mining, and social media monitoring. Sarcasm detection can help improve the accuracy of sentiment analysis by detecting sarcastic expressions that may be misclassified as positive or negative sentiment. Sarcasm detection can also help monitor social media for hate speech and cyberbullying.

FAQs

Q1. How is sarcasm different from irony?

Sarcasm is a type of irony that is expressed through language that is intended to convey the opposite of its literal meaning. Irony is a broader concept that includes various forms of speech that express a meaning that is different from the literal meaning.

Q2. Can sarcasm be positive?

Yes, sarcasm can be positive. While sarcasm is often associated with negative or mocking remarks, it can also be used in a positive way to convey humor, wit, or irony. Positive sarcasm can be a form of playful banter or teasing among friends or colleagues, and it can also be used to point out the absurdity of a situation or to make a point in a clever or humorous way.

For example, a teacher might use sarcasm to praise a student who has been absent frequently by saying "Oh, congratulations! You're finally here on a school day!" Or, a parent might use sarcasm to praise their child for making a mess while cooking by saying "Wow, you're quite the chef. You've even managed to get flour on the ceiling!"

In these examples, the sarcasm is used to convey a positive message in a humorous or lighthearted way. However, it's important to note that sarcasm can also be hurtful and insulting, depending on the context and tone in which it is used.

Q3. What is the most effective way for extracting characteristics from different categories, such as pragmatic and incongruity traits? Which feature category is more suited for sarcasm detection?

There are several methodologies to extract features related to various categories, including pragmatic and incongruity features, for sarcasm detection using deep learning. One popular approach is to use natural language processing (NLP) techniques to preprocess the text data and extract relevant features.

For example, for pragmatic features, NLP techniques can be used to extract information such as the speaker's attitude or intention, the context in which the statement is made, and the tone of the statement. Incongruity features, on the other hand, can be extracted by analyzing the disparity between what is expected and what is actually said in the text.

Both pragmatic and incongruity features are important for sarcasm detection, as they provide valuable cues to help distinguish between sarcastic and non-sarcastic statements. However, recent studies have shown that pragmatic features are generally more effective than incongruity features for sarcasm detection. This is because sarcastic statements often involve an ironic or insincere tone that is difficult to detect based solely on incongruity features.

In summary, the best methodology to extract features related to various categories for sarcasm detection using deep learning is to use NLP techniques to preprocess the text data and extract relevant features. Both pragmatic and incongruity features are important, but pragmatic features are generally more effective for sarcasm detection.

5.3 FUTURE WORK

Time is often a limited resource, and it can be challenging to explore every strategy or option when it comes to research or development. In the case of sarcasm detection, the authors of a recent paper noted that due to the shortage of time, they were unable to explore other strategies that they would like to focus on in the future. However, they did suggest some potential avenues for future research and development.

One such avenue is the use of FastText for word embedding instead of Word2Vec. The authors noted that all the papers they had read so far were using Word2Vec, but that FastText may provide better results. FastText is an extension of Word2Vec that was proposed by Facebook. It has some key benefits over Word2Vec, including its ability to work better on rare words. This is because FastText represents each word as a bag of character n-grams, which allows it to capture the sub-word information.

Another major benefit of using FastText is that it can handle Out-Of-Vocabulary (OOV) words which Word2Vec cannot. In other words, FastText can generate embeddings for words that are not present in the training corpus. This is particularly useful in situations where the language is constantly evolving, and new words or phrases are being introduced all the time. In addition to exploring different word embedding techniques, the authors also suggested that sarcasm detection could be extended to other languages. Specifically, they mentioned the possibility of detecting sarcasm in the Urdu language using an Urdu dataset. This could be an interesting area of research, as the nature of sarcasm may vary depending on the cultural context and linguistic features of different languages.

In general, the authors agreed that there was a wide range of prospective directions for further study and advancement in the area of sarcasm detection. Despite the time constraints, students were nevertheless able to provide some insightful recommendations for how this work may be improved and expanded in the future. It could be feasible to create more precise and trustworthy models that can be used in a variety of scenarios by continuing to investigate various methods and approaches for sarcasm detection.

REFERENCES

[1] [1] A comparison of the efficacy of sentiment tools and human coding in sarcasm detection, P. L. Teh, P. B. Ooi, N. N. Chan, and Y. K. Chuah, J. Syst. Inf. Technol., vol. 20, no. 3, pp. 358-374, 2018, doi: 10.1108/JSIT-12-2017-0120.

[2] A Pattern-Based Approach for Sarcasm Detection on Twitter, M. Bouazizi, T. O. Ohtsuki, and S. Member, IEEE Access, vol. 4, pp. 5477–5488, 2016, doi: 10.1109/ACCESS.2016.2594194.

[3] "Affective representations for sarcasm identification," 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018, pp. 1029–1032, 2018, doi: 10.1145/3209978.3210148. Agrawal and An.

[4] "Magnets for Sarcasm: Making Sarcasm Detection Timely, Contextual, and Very Personal," no. 2002, pp. 482-491, A. Ghosh and T. Veale, 2017.

[5] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," ACM Comput. Surv., vol. 50, no. 5, 2017, doi: 10.1145/3124420.

[6] D. Das and A. J. Clark, “Sarcasm detection on Flickr using a CNN,” ACM Int. Conf. Proceeding Ser., pp. 56–61, 2018, doi: 10.1145/3277104.3277118.

[7] D. Hazarika, S. Poria, S. Gorantla, E. Cambria, R. Zimmermann, and R. Mihalcea, “Cascade: Contextual sarcasm detection in online discussion forums,” arXiv, 2018.

[8] M. A. Ramdhani, M. A. Ramdhani, D. S. adillah Maylawati, and T. Mantoro, “Indonesian news classification using convolutional neural network,” Indones. J. Electr. Eng. Comput. Sci., vol. 19, no. 2, pp. 1000–1009, 2020, doi: 10.11591/ijeecs.v19.i2.pp1000-1009.

[9] D. I. H. Farías, “Irony and sarcasm detection in Twitter: The role of affective content,” Proces. Leng. Nat., vol. 62, no. September, pp. 107–110, 2019, doi: 10.26342/2019-62-14.

[10] K. Sundararajan and A. Palanisamy, “Multi-Rule Based Ensemble Feature Selection Model for Sarcasm,” vol. 2020, 2020.

[11] A. Kumar, V. T. Narapareddy, V. A. Srikanth, A. Malapati, and L. B. M. Neti, “Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM,” IEEE Access, vol. 8, pp. 6388–6397, 2020, doi: 10.1109/ACCESS.2019.2963630.

[12] S. Castro, D. Hazarika, V. Pérez-Rosas, R. Zimmermann, R. Mihalcea, and S. Poria, “Towards multimodal sarcasm detection (an obviously perfect paper),” ACL 2019 -

57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf., pp. 4619–4629, 2020, doi: 10.18653/v1/p19-1455.

[13] G. Abercrombie and D. Hovy, "Putting Sarcasm Detection into Context : The Effects of Class Imbalance and Manual Labelling on Supervised Machine Classification of Twitter Conversations," 2016.

[14] R. Misra and P. Arora, "Sarcasm Detection using Hybrid Neural Network," 2019, doi: 10.13140/RG.2.2.32427.39204.

[15] P. Rajapaksha, R. Farahbakhsh, and N. Crespi, "BERT, XLNet or RoBERTa: The Best Transfer Learning Model to Detect Clickbaits," IEEE Access, vol. 9, pp. 154704–154716, 2021, doi: 10.1109/ACCESS.2021.3128742.

[16] M. N. Asim, M. U. Ghani, M. A. Ibrahim, W. Mahmood, A. Dengel, and S. Ahmed, Benchmarking performance of machine and deep learning-based methodologies for Urdu text document classification, vol. 33, no. 11. Springer London, 2021.

[17] A. Aggarwal, A. Wadhawan, A. Chaudhary, and K. Maurya, "'Did you really mean what you said?' : Sarcasm Detection in Hindi-English Code-Mixed Data using Bilingual Word Embeddings," pp. 7–15, 2020, doi: 10.18653/v1/2020.wnut-1.2.

[18] L. H. Son, A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar, and M. Abdel-Basset, "Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network," IEEE Access, vol. 7, pp. 23319–23328, 2019, doi: 10.1109/ACCESS.2019.2899260.

[19] Y. Ren, D. Ji, and H. Ren, "Context-augmented convolutional neural networks for twitter sarcasm detection," Neurocomputing, vol. 308, pp. 1–7, 2018, doi: 10.1016/j.neucom.2018.03.047.

[20] S. Amir, B. C. Wallace, H. Lyu, P. Carvalho, and M. J. Silva, "Modelling context with user embeddings for sarcasm detection in social media," CoNLL 2016 - 20th SIGNLL Conf. Comput. Nat. Lang. Learn. Proc., pp. 167–177, 2016, doi: 10.18653/v1/k16-1017.

[21] Y. Yunitasari, A. Musdholifah, and A. K. Sari, "Sarcasm Detection For Sentiment Analysis in Indonesian Tweets," IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 13, no. 1, p. 53, 2019, doi: 10.22146/ijccs.41136.

[22] D. Ali, M. M. S. Missen, and M. Husnain, "Multiclass Event Classification from Text," Sci. Program., vol. 2021, no. 1, 2021, doi: 10.1155/2021/6660651.

[23] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," NAACL HLT 2019 - 2019

Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, no. Mlm, pp. 4171–4186, 2019.

[24] A. Joshi, P. Bhattacharyya, M. Carman, J. Saraswati, and R. Shukla, "How do cultural differences impact the quality of sarcasm annotation?: A case study of Indian annotators and American text," in Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, 2016, pp. 95-99.

[25] A. Joshi, P. Bhattacharyya, and M. J. Carman, "Automatic sarcasm detection: A survey," ACM Computing Surveys (CSUR), vol. 50, no. 5, pp. 1-22, 2017.

[26] S. Kannangara, "Mining twitter for fine-grained political opinion polarity classification, ideology detection, and sarcasm detection," in Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, 2018, pp. 751-752.

[27] B. Liu, "Sentiment analysis and subjectivity," Handbook of Natural Language Processing, vol. 2, pp. 627-666, 2010.

[28] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm detection on Twitter: A behavioral modeling approach," in Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, 2015, pp. 97-106.

[29] S. M. Sarsam, "Reinforcing the decision-making process in chemometrics: Feature selection and algorithm optimization," in Proceedings of the 2019 8th International Conference on Software and Computer Applications, 2019, pp. 11-16.

[30] D. G. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis," in LREC 2014 Proceedings. ELRA, 2014, March.

[31] D. I. H. Farías, V. Patti, and P. Rosso, "Irony detection in Twitter: The role of affective content," ACM Transactions on Internet Technology (TOIT), vol. 16, no. 3, pp. 1-24, 2016.

[32] R. González-Ibánez, S. Muresan, and N. Wacholder, "Identifying sarcasm in Twitter: a closer look," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, June, pp. 581-586.

[33] O. Tsur, D. Davidov, and A. Rappoport, "ICWSM—A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews," in Fourth International AAAI Conference on Weblogs and Social Media, 2010, May.

[34] R. Kreuz and G. Caucci, "Lexical influences on the perception of sarcasm," in Proceedings of the Workshop on Computational Approaches to Figurative Language, Apr. 2007, pp. 1-4.

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

[36] B. C. Wallace, L. Kertz, and E. Charniak, "Humans require context to infer ironic intent (so computers probably do, too)," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Jun. 2014, pp. 512-516.

[37] D. Davidov, O. Tsur, and A. Rappoport, "Semi-supervised recognition of sarcasm in Twitter and Amazon," in Proceedings of the Fourteenth Conference on Computational Natural Language Learning, Jul. 2010, pp. 107-116.

[38] P. Liu, W. Chen, G. Ou, T. Wang, D. Yang, and K. Lei, "Sarcasm detection in social media based on imbalanced classification," in Web-Age Information Management: 15th International Conference, WAIM 2014, Macau, China, June 16-18, 2014. Proceedings 15, Springer International Publishing, 2014, pp. 459-471.

[39] O. Tsur, D. Davidov, and A. Rappoport, "ICWSM—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews," in Fourth International AAAI Conference on Weblogs and Social Media, May 2010.

[40] R. González-Ibánez, S. Muresan, and N. Wacholder, "Identifying sarcasm in Twitter: a closer look," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Jun. 2011, pp. 581-586.

[41] C. Chung and J. W. Pennebaker, "The psychological functions of function words," Social Communication, vol. 1, pp. 343-359, 2007.

[42] A. Reyes, P. Rosso, and D. Buscaldi, "From humor recognition to irony detection: The figurative language of social media," Data & Knowledge Engineering, vol. 74, pp. 1-12, 2012.

[43] D. G. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis," in LREC 2014 Proceedings, ELRA, Mar. 2014.

[44] Tepperman, J., Traum, D., & Narayanan, S. (2006). " Yeah right": sarcasm recognition for spoken dialogue systems. UNIVERSITY OF SOUTHERN CALIFORNIA LOS ANGELES.

[45] Tsur, O., Davidov, D., & Rappoport, A. (2010, May). ICWSM—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In fourth international AAAI conference on weblogs and social media.

[46] Carvalho, P., Sarmento, L., Silva, M. J., & De Oliveira, E. (2009, November). Clues for detecting irony in user-generated contents: oh...!! it's" so easy";-. In Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion (pp. 53-56).

[47] Davidov, D., Tsur, O., & Rappoport, A. (2010, July). Semi-supervised recognition of sarcasm in Twitter and Amazon. In Proceedings of the fourteenth conference on computational natural language learning (pp. 107-116).

[48] González-Ibánez, R., Muresan, S., & Wacholder, N. (2011, June). Identifying sarcasm in twitter: a closer look. In Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies (pp. 581-586).

[49] Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., & Huang, R. (2013, October). 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).

[50] Joshi, A., Sharma, V., & Bhattacharyya, P. (2015, July). Harnessing context incongruity for sarcasm detection. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers) (pp. 757-762).

[51] Lukin, S., & Walker, M. (2017). Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. arXiv preprint arXiv:1708.08572.

[52] Liebrecht, C. C., Kunneman, F. A., & van Den Bosch, A. P. J. (2013). The perfect solution for detecting sarcasm in tweets# not.

[53] Ptáček, T., Habernal, I., & Hong, J. (2014, August). Sarcasm detection on czech and english twitter. In Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers (pp. 213-223).

[54] Kreuz, R., & Caucci, G. (2007, April). Lexical influences on the perception of sarcasm. In Proceedings of the Workshop on computational approaches to Figurative Language (pp. 1-4).

[55] Wallace, B. C., Kertz, L., & Charniak, E. (2014, June). Humans require context to infer ironic intent (so computers probably do, too). In Proceedings of the 52nd Annual

Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 512-516).

[56] Rajadesingan, A., Zafarani, R., & Liu, H. (2015, February). Sarcasm detection on twitter: A behavioral modeling approach. In Proceedings of the eighth ACM international conference on web search and data mining (pp. 97-106).

[57] Bamman, D., & Smith, N. (2015). Contextualized sarcasm detection on twitter. In proceedings of the international AAAI conference on web and social media (Vol. 9, No. 1, pp. 574-577).

[58] Khattri, A., Joshi, A., Bhattacharyya, P., & Carman, M. (2015, September). Your sentiment precedes you: Using an author's historical tweets to predict sarcasm. In Proceedings of the 6th workshop on computational approaches to subjectivity, sentiment and social media analysis (pp. 25-30).

[59] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, et al., "Large batch optimization for deep learning: Training bert in 76 minutes," arXiv preprint arXiv:1904.00962, 2019.

[60] R. K. Kaliyar, A. Goswami, and P. Narang, "FakeBERT: Fake news detection in social media with a BERT-based deep learning approach," Multimedia Tools and Applications, vol. 80, no. 8, pp. 11765-11788, 2021.

[61] H. S. Alatawi, A. M. Alhothali, and K. M. Moria, "Detecting white supremacist hate speech using domain specific word embedding with deep learning and BERT," IEEE Access, vol. 9, pp. 106363-106374, 2021.

[62] C. I. Eke, A. A. Norman, and L. Shuib, "Context-based feature technique for sarcasm identification in benchmark datasets using deep learning and BERT model," IEEE Access, vol. 9, pp. 48501-48518, 2021.

[63] Y. Liu, J. Lu, J. Yang, and F. Mao, "Sentiment analysis for e-commerce product reviews by deep learning model of Bert-BiGRU-Softmax," Mathematical Biosciences and Engineering, vol. 17, no. 6, pp. 7819-7837, 2020.

[64] M. R. Islam, A. R. M. Kamal, N. Sultana, R. Islam, and M. A. Moni, "Detecting depression using k-nearest neighbors (knn) classification technique," in 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), Feb. 2018, pp. 1-4.

[65] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in Advances in neural information processing systems, vol. 28, 2015.

[66] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5, 2011, pp. 507-523.

[67] R. S. Olson, R. J. Urbanowicz, P. C. Andrews, N. A. Lavender, L. C. Kidd, and J. H. Moore, "Automating biomedical data science through tree-based pipeline optimization," in Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30--April 1, 2016, Proceedings, Part I, 2016, pp. 123-137.

[68] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-WEKA: Automatic model selection and hyperparameter optimization in WEKA," in Automated machine learning: methods, systems, challenges, 2019, pp. 81-95.

[69] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, July 2019, pp. 1946-1956.

[70] S. Umemura, H. Arima, S. Arima, K. Asayama, Y. Dohi, Y. Hirooka, et al., "The Japanese Society of Hypertension guidelines for the management of hypertension (JSH 2019)," Hypertension Research, vol. 42, no. 9, pp. 1235-1481, 2019.

[71] Y. F. Li, H. Wang, T. Wei, and W. W. Tu, "Towards automated semi-supervised learning," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, July 2019, pp. 4237-4244.

[72] D. Howard, M. M. Maslej, J. Lee, J. Ritchie, G. Woollard, and L. French, "Transfer learning for risk classification of social media posts: Model evaluation study," Journal of medical Internet research, vol. 22, no. 5, May 2020, e15371.

[73] Z. Chen, H. Zhang, X. Zhang, and L. Zhao, "Quora question pairs," 2018.

[74] Jiang, B., & Wang, W. (2019). Auto-weighted group lasso for multi-task learning with application to functional connectome data analysis. Journal of Neuroscience Methods, 326, 108363.

[75] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

[76] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[77] Word Tweet Cloud Bot. [Online]. Available: https://twitter.com/wordnuvola. [Accessed: Feb. 10, 2023].

[78] Mentionlytics. (2023). Intelligent Social Media Monitoring. [Online]. Available: https://www.mentionlytics.com/. [Accessed: Feb. 11, 2023].