DESIGN OF A FRAMEWORK FOR DETECTION OF IMPLICIT AND SARCASM REVIEWS

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Applications

By

Parkar Ameya Keshav

Registration Number: 42100054

Supervised By

Dr. Rajni Bhalla (UID: 12936)

Professor, School of Computer Applications

Lovely Professional University

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB
2025

DECLARATION

I, hereby declared that the presented work in the thesis entitled "**DESIGN OF A FRAMEWORK FOR DETECTION OF IMPLICIT AND SARCASM REVIEWS**" in fulfillment of degree of **Doctor of Philosophy (Ph. D.)** is outcome of research work carried out by me under the supervision of Dr. Rajni Bhalla, working as Professor, in the School of Computer Applications of Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgements have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

(Signature of Scholar)

Name of the scholar: Ameya Parkar

Registration No.: 42100054

Department/School: School of Computer Applications

Lovely Professional University,

Punjab, India

CERTIFICATE

This is to certify that the work reported in the Ph. D. thesis entitled **DESIGN OF A FRAMEWORK FOR DETECTION OF IMPLICIT AND SARCASM REVIEWS**" submitted in fulfillment of the requirement for the award of degree of **Doctor of Philosophy (Ph.D.)** in the School of Computer Applications, is a research work carried out by Ameya Parkar, 42100054, is bonafide record of his/her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.

(Signature of Supervisor)

Name of supervisor: Dr. Rajni Bhalla

Designation: Professor

Department/school: School of Computer Applications

University: Lovely Professional University

ABSTRACT

From the past two decades, social media has led to a lot of information online in the form of text, audio, video, etc. Textual data occupies a large chunk of the data found online. People tend to post about things they see, things they like, etc. on different online platforms. People tend to post about their feelings, about their opinions on different products such as mobile phones, cameras, etc. Understanding the opinion of people is important for businesses and organizations to gauge about their products and services.

A lot of popular websites and apps like Flipkart, Instagram, etc. are accessed by different groups of people online. People tend to post their likings, their choices, their preferences on such websites and apps. The websites allow the posts to be textual, in voice and video formats. Due to this reason, a humongous amount of data is posted online and processing of such data in raw format can be quite tedious. Institutes, businesses, etc. can use this data for extracting important information about their products and services. Natural language processing (NLP) offers us multiple options like opinion mining, detection of emotions, extraction of aspects, etc. These techniques can be used by institutes and companies to understand the different aspects of their products and services. Depending on the techniques used and the outcome of the techniques, vital information can be gathered and actions to be taken from a future point of view can be determined. Polarity mining can lead to understand the fundamental views of the products and services of the general public.

NLP is a subset of AI & one of the most important tasks in today's world. Different tasks such as prediction, sentiment detection, aspect detection, sarcasm detection, translation from one language to another, emotion detection, etc. fall under Natural Language processing.

There are many sub categorical domains in opinion mining. In emotion detection, the emotion of the reviewer is found out. For example, after watching a movie, a person might write a review depicting exciting emotion. Another person showed nostalgia as the

emotion for the same movie. In aspect extraction, features/aspects of an entity are picked and their sentiments are expressed. For example, after the launch of a new mobile phone, the reviewer writes about different aspects of the mobile separately. The person expresses differing opinions on the different aspects so as to judge the aspects individually rather than as a whole entity (phone).

Nowadays automation techniques have been introduced in the different domains of natural language processing. The models automatically detect the sentiment and the output is found out. But, in some scenarios, automation might not be the best choice. In websites and applications like “X”, there are a lot of tweets and links along with different emoticons. Also, the language used is a shortcut language rather than using the whole grammatically correct sentence. Also, for automation, large data size is required for good performance which may not be the case for all domains.

Automation is difficult in tasks such as presence of sarcasm in text. Detection of sarcasm is another complex task in Natural language processing. People tend to post their opinions in a positive way in spite of the opinion being negative. This could trick any sentiment analysis model to rate the review as positive in spite of being negative.

Sometimes, people tend to be sarcastic in their opinions. Due to sarcasm, the polarity of a particular opinion can be reversed and a different picture is portrayed than what it originally is. Hence, it is necessary to detect sarcastic posts and classify them properly. A lot of work is done on detecting sarcasm using rule based methods, classification methods of machine learning, etc. Lately, the focus has been to use deep learning to detect sarcasm.

Another thing to factor in the opinions of people is that people nowadays express about different details of an entity/product. In the case of mobile phones, people tend to talk about the camera, memory, speed, video quality, charging, etc. So a particular aspect of the mobile phone could be positive while some other aspect could be negative. Hence, it is necessary to take opinions aspect wise about the product. Some aspects are clearly

mentioned in the opinion while some are indirectly referred to. Features can be categorized as implicit or explicit. The explicit aspects are clearly mentioned in the review. The indirectly referred aspects, also known as implicit aspects, need to be identified also. They are indirectly referred to in the review. In aspect detection, recognizing the implicit aspects is very important as the owner of the review might be describing different opinions on different aspects of mobiles. Some opinions of certain aspects could be good but some might not.

This thesis focuses on detecting sarcasm as well as finding implicit aspects in mobile reviews. We have used hybrid framework in deep learning to detect sarcasm while encoder-decoder technique is used to detect implicit aspects. We have used 3 datasets for the same. While applying deep learning techniques for sarcasm, we used techniques like RNN, LSTM, etc. and found that the hybrid framework gives the best performance. For implicit aspect detection, we used techniques such as co-occurrence matrix, rule-based methods and encoder-decoder. The novelty of this research is that in the domain of mobile reviews, encoder decoder technique has been used in conjunction with supervised learning as a backup.

The encoder-decoder technique gave the best performance to detect implicit aspects. The performance was judged on different performance metrics such as accuracy, precision, recall and f-score. The frameworks we have designed will definitely benefit businesses and organizations in terms of detection of sarcasm and getting to know the implicit aspects. Our frameworks in turn will benefit other researchers and students to take this work ahead and get better performances in the nearby future.

The outcomes for this thesis are:

1. A customized dataset has been made in the domain of mobile reviews by gathering data from a questionnaire as well as scrapping data from Twitter website.
2. A hybrid framework has been made to detect the presence of sarcasm in text reviews.
3. A novel framework has been made to find the implicit aspects in text reviews.

ACKNOWLEDGEMENT

I thank the Almighty for his blessing in making this endeavor a success. Completion of this doctoral thesis was possible with the help of many people, and I would like to express my gratitude to all of them.

I express my deep sense of thanks to my supervisor Dr. Rajni Bhalla, Professor, School of Computer Applications, Lovely Professional University, Punjab.

She has continuously motivated me by providing scholarly inputs with timely critical comments and unconditional support for this PhD work. Without her support, this thesis might not have existed.

I express my deep sense of thanks to my chancellor, Dr. Ashok Mittal, LPU for providing me an opportunity to carry out my research work in this university.

I wish to thank the members of the Doctoral Committee for their invaluable suggestions.

This research might have remained just a dream, and I may not have reached this position without the blessings of my father (Mr. Keshav Parkar) and mother (Mrs. Ketaki Parkar).

I also thank my wife (Mrs. Priyanka Parkar) and my children (Master. Anvay Parkar & Master. Advay Parkar) for being a major moral factor in helping me reach this stage. I am grateful to them for their continuous motivation and support which is irreplaceable.

Further, I thank all my well-wishers for their kindness and cooperation.

AMEYA PARKAR

TABLE OF CONTENTS

List of Tables

List of Figures

List of Appendices

Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
NB	Naïve Bayes
SVM	Support Vector Machine
RF	Random Forest
LR	Logistic Regression
GB	Gradient Boost
DT	Decision Tree
kNN	k nearest neighbor
SGD	Stochastic Gradient Descent
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
CNN	Convolution Neural Network
GloVe	Global Vectors for Word Representation
BiLSTM	Bi-directional LSTM
BERT	Bidirectional Encoder Representations from Transformers
A	Accuracy
P	Precision
R	Recall
F	F score
CB	Hybrid framework
EDS	Encoder Decoder framework with supervised learning as backup

Chapter 1
INTRODUCTION

1.1 INTRODUCTION

Sentiment Analysis/Opinion Mining is a technique used to find the opinions/sentiments of different topics. In a general way, sentiment analysis/opinion mining tells us the opinion of a particular entity/topic as positive, negative or neutral. However, it can be split up further in terms of numbering or in terms of stars, etc.

Opinion mining techniques are done from different perspectives. Majorly, it is carried out on the whole review, partial review or aspect level. An aspect/feature describes about a particular thing about the entity/product. A mobile phone can have different aspects such as camera, design, speed, memory, etc. A person could be speaking about any of the aspects of a mobile phone where one aspect could be positive while one aspect could be negative.

Sarcasm detection is a sub category of sentiment analysis. Sarcasm in its literal meaning suggests that a person is expressing an opinion having inverse polarity. i.e. the person could be expressing an opinion in a positive way but it is negative. The opinion could be negatively said but could be positive. But most of the cases are found where the person portrays his/her opinion in a positive way but is actually negative. A simple comment such as “I like to be ignored” can be termed as positive by a computer model because of the presence of the word “like”. However, it is very clear to a human mind after looking at the entire comment that it is negative.

An aspect could be explicit or implicit. If the aspect is clearly mentioned in the opinion, then it is categorized as an explicit opinion. However, if it is not mentioned in the opinion and the opinion is implicitly referring to it, it is categorized as an implicit opinion. In the opinion “My phone is huge in size but light”. Here, the aspect “size” is clearly mentioned in the review and it is categorized as an explicit aspect. However, the person is also

expressing about the phone being light which implies that the person is talking about the aspect “weight” of the phone. Since, the aspect “weight” is not mentioned in the opinion, it is categorized as an implicit aspect.

Designing a framework for Sarcasm detection and another framework for Implicit Aspect detection is the primary goal of this study.

1.1.1 Sentiment Analysis

Opinion Mining is the step by step method of determining if a particular review/sentence/message/document is positive, negative or neutral. A lot of data is generated online from chats, social media, reviews of products, political discussions, etc. By performing sentiment analysis, we can judge the opinion of the person about the topic. Institutes/companies/organizations/people can use this analysis to determine the popularity about the topic and then decide how to go ahead with their strategies w.r.t. brand reputation, service, etc.

Sentiment analysis gives a deep insight about the opinions of customers. Sometimes, people can only take the positive part of the review and discard the negative part. By using sentiment analysis, we can actually consider the whole opinion and ensure that there is no bias. Sentiment Analysis allows companies to know about their products opinions, feedback, etc. The amount of data generated is humongous and by using sentiment analysis, the data can be analyzed and in a short interval of time it can be determined if the opinions are positive, negative or neutral. Real time data can be scrapped or extracted and immediate actions can be carried out depending on the results.

There are multiple use cases of sentiment analysis. Companies use the data to analyze and prepare different plans for the present and the future. Customer service uses this data to personalize responses and can decide which components are important and should be prioritized. Companies can learn about their products and check what offerings work and what don’t. The recommendations are then passed to the technical and management team to innovate and make changes if necessary. Some companies have advertising campaigns

and the results of such advertising campaigns are shared with the top personnel of the company to decide on future plans.

1.1.2 Steps to do Sentiment Analysis

1. **Collection of data** from various sources.
    a. Datasets could be publicly available on websites such as Kaggle, Github, etc.
    b. Dataset can be made by scrapping data from websites such as Twitter, Reddit, Amazon, Flipkart, etc.
    c. Real time data can also be downloaded by scrapping from Twitter, etc.
    d. A customized dataset can be created by gathering responses online, using a questionnaire, offline from places such as malls, companies, etc.
2. **Preprocessing** of data. Once the dataset is ready, preprocessing techniques can be applied on the dataset.
    a. Tokenization can be done to split every sentence in every review into multiple individual parts.
    b. Stop words are removed from the dataset.. Words such as is, the, have, etc. are examples of stop words.
    c. Stemming/Lemmatization are done on the dataset to convert words to their true form. For example, words like systematic are converted to its basic word which is system.
3. **Word embeddings**. The content is modified to vector format by using different word embeddings. Bag of Words, Term frequency Inverse Document frequency, Word2Vec, FastText, GLOVE, etc. are used to convert the content to vectors.
4. The dataset is split into **training and testing** in ratio such as 80:20 or 70:30 depending on the type of dataset and its context. Programmers may also keep data for validation from the preprocessed dataset. Fold cross validation can also be used at this stage.

5. **Analysis** of data. Techniques such as machine learning, deep learning, rule based methods, co-occurrence matrix, topic modeling, etc. are used on the dataset to train the model and make it ready for prediction.
6. **Testing**. The model is tested on the testing dataset and the review/sentence is classified as positive, negative or neutral.
7. **Performance metrics**. The model is judged on different performance metrics such as accuracy, precision, recall and f-score.

1.1.3 Different types of Sentiment Analysis

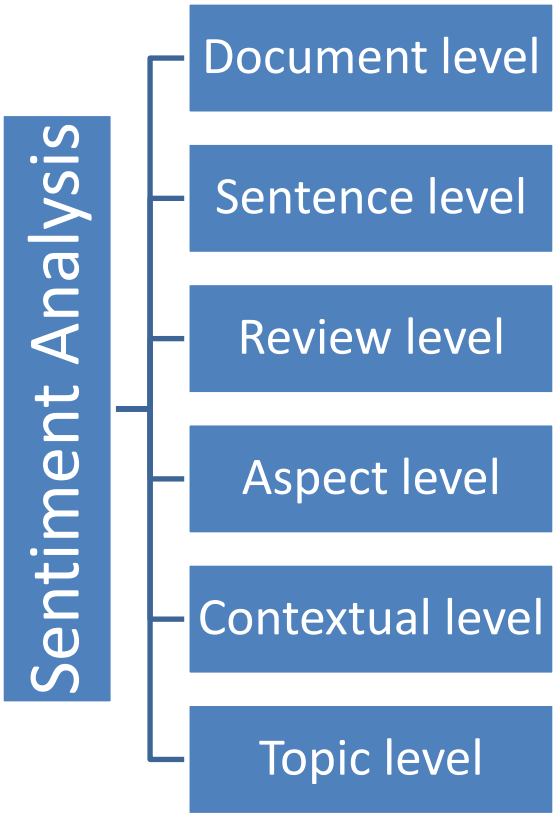

Figure 1.1 Types of Sentiment Analysis

Sentiment Analysis can be of different types:

1. **Document level Sentiment Analysis**: In document level sentiment analysis, the entire document as a whole is checked for polarity. It goes with the assumption that the entire document is dedicated for a single entity.
2. **Sentence level Sentiment Analysis**: In sentence level sentiment analysis, a sentence is picked from a document/review and the polarity is determined for the sentence.
3. **Review level Sentiment Analysis**: In review level sentiment analysis, an entire review is picked and polarity is determined. It is similar to sentence level sentiment analysis with the difference been that the review may contain more than one sentence.
4. **Aspect level Sentiment Analysis**: In aspect based sentiment analysis, aspects are extracted from document/sentence/review and the polarity is found for each individual aspect.
5. **Contextual level Sentiment Analysis**: In contextual level sentiment analysis, clues are picked up from the review/sentence and the context is checked w.r.t. how the meaning of words changes with context. The opinion is dependent on the way the words are written in the sentence/review.
6. **Topic level Sentiment Analysis**: In topic level sentiment analysis, important topics relevant to the task being done are extracted and the opinion about each topic is found out and polarity is determined.

1.1.4 Applications of Sentiment Analysis

Figure 1.2 Applications of Sentiment Analysis

1. **Market Research**: Sentiment Analysis is used to judge the polarity of the entire market, different segments in the market, specific products, etc. It helps decision makers reach to a conclusion in the decision making process.
2. **Brand Reputation**: Companies advertise and sell their products in online/offline mode. By gathering the opinions of different sections of society and finding the polarity about their product(s), the companies can determine the future course of actions and help creating a brand for new companies as well as maintaining a brand for existing companies.
3. **Service Improvement**: Service oriented companies like Ola, Uber, etc. rely on the opinions of people and through social media gather information and find the polarity about their services w.r.t. different aspects. An example could be the opinions of people on their newly launched app or latest version of their app.
4. **Movie Reviews**: Companies like IMDB, Rotten Tomatoes, etc. use sentiment analysis to analyze movie reviews. Opinions of different people including experts are gathered and the sentiment is found out. They give a numerical rating out of 5 or 10 and some companies classify them in different categories such as Hit, Super Hit, Neutral, Flop, etc. Some companies rate the movie reviews by allocating stars to the movie performance.
5. **Politics**: Different political parties take the sentiments of people and through exit polls determine the chances of which political party will be winning the election. If negative sentiments are found out about a particular party, then the party can work on the negative points and see if they can improve on their performance and win the upcoming election.
6. **Customer Support**: People dislike a particular product or brand on the type of customer support offered by a company/brand. With the advent of social media, many people post their reviews online on different social media companies and that in turn, may adversely affect a business.

7. **Employee Satisfaction**: Using sentiment analysis, an organization can understand the productivity and the outlook of the employees in the organization. Employee feedback can be taken and steps to improve employee satisfaction can take place.
8. **Finance**: The stock market is volatile in nature and fluctuation happens quite often on events happening related to the company. Sentiment analysis can be done by taking opinions of experts and in general public to gauge the trend and make decisions accordingly.
9. **Improving products**: Employees can take the sentiment analysis report for a particular product and can think of ways to improve their product. Marketing campaigns could be held depending on the results and the product(s) could be improved.

1.1.5 Challenges faced in Sentiment Analysis

1. **Tone**: The tone in which the reviewer says could be deceiving and it would be more difficult to figure it in written text. Just by looking at a written review, it would be difficult to judge the tone and the correct sentiment of the reviewer. For example, “This book is good but not at that price”.
2. **Polarity**: The presence of positive words and negative words can change the polarity. If the sentence has words such as “good”, “better”, “bad”, then there are 2 positive sentiments and one negative which would indicate that the polarity is positive. Although, that may not be the case always.
3. **Sarcasm**: In casual conversations, people tend to use sarcasm. A text is said to be sarcastic if it seems positive but in fact it is negative and vice-versa. It becomes difficult for a model to detect sarcasm. Detection of sarcasm is a need and by doing so, sentiment analysis could be done properly.
4. **Implicit Aspects**: In aspect based sentiment analysis, polarity is found out for each aspect individually. Aspects which are mentioned in the review are called as

explicit aspects while those missing in the review are called as implicit aspects. Many users express their views without mentioning the aspect.

5. **Emojis**: In websites like X, Instagram, etc. , people express emojis in abundance. The sentiment analysis models do not understand the emojis. Many researchers may remove the emojis as they consider them as special characters and they are filtered out in the pre-processing steps. The removal of emojis could in turn flip the polarity of the review.
6. **Idioms**: Computer trained models generally do not understand idioms. An idiom like "add fuel to the fire" stands for a situation getting worse but could be interpreted differently by a model.
7. **Negation**: Multiple negations in a review can confuse a computer model. A review "I definitely won't want to not miss today's football practice" can be interpreted negative by a model when in fact it is positive.
8. **Comparisons**: Comparison of two objects/entities normally does not carry an opinion and it has to found out. For example, "The Mercedes car is more reliable than the Renault car". The review does not have any sentiment and just carries an opinion of comparison between the two cars.
9. **Employee bias**: Feedback of employees is certainly valuable to judge about the inner workings of the company. Many companies have certain biases and outlooks which do not keep the employee feedback as important as customer feedback.
10. **Multilingualism**: Many reviews are expressed nowadays which have a mixture of languages. For every language, a different POS tagger, lemmatizer, grammar, etc. are needed. For example, a phrase in English might have a different meaning if converted to another language.

1.1.6 Sarcasm

People express their views online on different platforms including social media, shopping websites, etc. Some of them express their true opinions on an entity/product while some

express it sarcastically. A sarcastic review actually means the opposite of what is being expressed by the person. A positive sentiment post can actually be negative due to the presence of sarcasm and vice-versa. For example, “This phone is so awesome that it lags all the time”. Here, the person is expressing sarcasm by portraying a negative opinion in a positive way.

People try to express their opinions sarcastically when they are sad, angry, depressed, frustrated, etc. Instead of directly mentioning these feelings in the review, then express it sarcastically. Some people use sarcasm to add humor and make the readers of the reviews laugh.

Sarcasm is a major challenge in sentiment analysis as computer models will find it difficult to detect and sentiment could be opposite than what is being portrayed.

1.1.6.1 Types of Sarcasm

1. **Self-deprecating sarcasm**: When the reviewer ridicules himself in the review, it is called as self-deprecating sarcasm. For example, “I am so good in Mathematics that I scored 20% in the exam”.
2. **Pity sarcasm**: When the reviewer expresses pity for himself/herself in a situation, it is called as pity sarcasm. For example, if a salaried person is asked to work beyond his normal working hours and says “Wonderful!!! Some more extra hours of work with no extra money”.
3. **Expressionless sarcasm**: When the reviewer expresses an opinion with no emotions, it is called as expressionless sarcasm. For example, when a person says “Yay!! I am looking forward to the football match tonight” with a dead expression face. It would be difficult to check if the person wants to watch the match or not.
4. **Polite sarcasm**: When the reviewer gives a polite opinion but is meant to be not sincere, it is called as polite sarcasm. For example, “Your house is so beautiful” but does not mean it.

5. **Offensive sarcasm**: When a person puts an offensive remark to offend another person, it is called as offensive sarcasm. For example, “What a party! It was boring till the last moment”.
6. **Anger sarcasm**: When a reviewer uses anger to express his/her opinion, it is called as anger sarcasm. For example, “What an idea!!! Let me work all day in office and then come home and work at home!!”

1.1.7 Implicit Aspects

In a recently launched mobile phone, the reviewers posted in detail about different aspects of the phone such as display, camera, speakers, selfie camera, ultra wide angle camera, battery life, charging time, etc. The people watching these videos or reading about it on social media websites get influenced. Also, many people are interested in specific features of the phone such as camera, dust resistance rating, under water rating, etc. and depending on the expert opinions of the reviewers on different aspects, the people decide to buy or not to buy the mobile phone. Each and every aspect of the phone is explained by these reviewers, which makes people realize and understand the significance of each and every aspect of the mobile phone.

Keeping these things in mind, we decided to focus on aspect detection for mobile phones. Also, people tend to write reviews in a way that the feature(s) is not directly mentioned in the review. Detecting these aspects for a normal person can be difficult as well as for a machine. For example, the mobile review is “My phone of XYZ Company has an outstanding back camera but not so good selfie camera. It is smooth and doesn’t lag”. Here, the aspect “camera” is clearly mentioned in the review and hence, it is an explicit aspect. However, the mobile being smooth (aspect: touch) and lag (aspect: usability) are not directly mentioned in the review. Hence, touch and usability are implicit aspects. Our study is focused on finding implicit aspects in the mobile reviews.

1.2 Languages and Tools for programming in this study

1.2.1 Python

The tasks of Sarcasm Detection and Implicit Aspect Detection fall under Natural Language Processing. Python gives many modules like NLTK, spaCy, etc. which have inbuilt functions to make the task of NLP easier. The following tasks can be done using the different libraries of Python.

1. Data collection using scrapping by using libraries such as tweepy, snscrape, beautifulsoup, etc.
2. Pre-processing dataset by using tokenization, removal of stop words, remove unwanted symbols, lemmatization, stemming, etc.
3. Text analysis.
4. Use different functions to do machine learning and deep learning tasks.
5. Create visualizations to analyze the performance such as plots, graphs, etc.

1.2.2 **Google Colab**

Google colab is an online notebook environment that allows users/programmers to train and test different ML and DL algorithms using multiple GPUs/TPUs. The advantage is that it could be used on any machine with any configuration. The advantages are:

1. **Convenience**: It can be accessed from any browser on any platform. Also, there is no need to install any software on the computer. Only Internet is required to access it.
2. **Strength**: GPUs and TPUs are available in Google Colab for faster processing & powerful resources.
3. **Sharing**: Google Colab notebooks can be shared with other users/programmers. Editing and running of the code can be done in real time.

4. **Libraries and Documentation**: Google colab offers different tools for learning about different aspects of Python.
5. **Free**: Google colab is free for a certain amount of memory. If more memory is required then there are plans available on a chargeable basis to use.

1.2.3 **Jupyter Notebook**

Jupyter Notebook is a notebook available in the open source software of Anaconda. It is used to do programming for ML and DL techniques. It uses the RAM of the device where the notebook is installed. The advantages of using Jupyter notebook are:

1. **Interactive**: Users can try different codes in the interactive panel and get instant results.
2. **Visualization**: Different graphs and tables can be created using simple commands for real time data.
3. **Communication**: Jupyter Notebook can be used to create and show live data and analysis and communicate it to clients/users.
4. **Education**: Jupyter Notebook can be used for educational purposes and as a learning tool.

1.3 Title of the proposed research work:

Design of a framework for detection of implicit and sarcasm reviews

1.4 Research Gaps

1. Sarcasm is expressed in different ways by different people in text reviews. With advance in technology, an updated system to detect sarcasm could be used to find sarcasm in text.
2. Many researchers keep the explicit features in mind but ignore the implicit features which could mean loss of crucial information about the entity sentiment wise. Researches are ongoing to detect not only explicit but also implicit features.

However, a lot of work can be done to improve the current results in the field.

1.5 Objectives of the proposed research work:

1. Collection of data using online sources as well as datasets used by other researchers from Twitter.
2. Comparative study of existing models for implicit and sarcasm detection.
3. To develop a framework for prediction using hybrid approaches for sarcasm detection.
4. To develop a novel framework to detect implicit aspects.
5. To evaluate the framework and check validity for prediction and monitor performance metrics.

Chapter 2

REVIEW OF LITERATURE

The following is the literature review for the study.

2.1 Proposed Methodology for Achievement of the Objectives

Table 2.1: Methodology/ Tools to be used

Objective	**Analysis to be under taken**	**Software to be used**	**In house availability (Yes/No)**	**Institute where facility is available**
Collection of data using online sources as well as datasets used by other researchers for Twitter.	Data collection using online sources and questionnaire	Google colab using Python, Google form for questionnaire and physical reviews collected	Yes	Not applicable
Comparative study of existing models for implicit and sarcasm detection	Comparison of existing techniques	Google colab using python	Yes	Not applicable
To develop a framework for prediction using hybrid	Hybrid approach to detect sarcasm	Google colab using python	Yes	Not applicable

approaches for sarcasm detection				
To develop a novel framework to detect implicit aspects	Novel approach to detect implicit aspects	Google colab using python	Yes	Not applicable
To evaluate the framework and check validity for prediction and monitor performance metrics	Comparison of proposed framework with existing frameworks	Google colab using python	Yes	Not applicable

2.2 Literature review for Sentiment Analysis of Mobile reviews

Bhalla R. et al. [1] researched on the problem of frequency of any class/event being zero in Naïve Bayes algorithm. They resolved the issue by proposing a new RB-Bayes method.

Bhalla R. et al. [2] made a comparison study of existing machine language models on mobile reviews and found out that Naïve Bayes gave the best performance for their dataset.

Nahili W. et al. [3] took their dataset of mobile reviews from Amazon to classify the reviews in 5 categories: very negative, negative, neutral, positive and very positive. Python was the language and VADER was the analysis tool used.

Minu P. et al. [4] used machine learning classifiers for opinion mining on mobile phone dataset from Kaggle website. The dataset contained data from 5 companies and classification was done using algorithms such as KNN, Naïve Bayes and SVM.

Rajkumar J. et al. [5] researched on sentiment analysis by downloading an Amazon dataset of various products including mobile phones and used SVM & Naïve Bayes classifiers.

Yashaswini H. et al. [6] worked on finding sentiments of mobile reviews written in Kannada language. Naïve Bayes classifier was used to find the sentiments.

Zeenia S. et al. [7] used SVM classifier using cross validation to detect sentiments of reviews collected from Amazon.

Salem M. et al. [8] used entropy method and Naïve Bayes method on mobile reviews dataset extracted from Kaggle website.

Fangyu W. et al. [9] used SenBERT CNN technique for Chinese reviews on mobiles. BERT was used for embeddings and CNN to extract the text aspects.

Pandey P. et al. [10] used POS tagging to find out adjectives, verbs, nouns, adverbs, etc. in the reviews individually and used it to detect the sentiment of the reviews.

Seema Y. et al. [11] used a dictionary approach to detect the sentiments of reviews on restaurants, mobiles, movies, etc.

Singh W. et al. [12] gathered opinions of mobiles from shopping sites. They found sentiments of different aspects of mobile phones such as camera, video, etc.

Table 2.2 Related work on Sentiment Analysis for mobile reviews

References	**Objective of the paper**	**Dataset Size**	**Language**	**Performance A:Accuracy P:Precision R: Recall F: F-score**
[3]	Finding sentiment of Amazon reviews	Amazon 400000 reviews	English	Statistical Analysis A: 76.80
[4]	ML techniques for Sentiment Analysis of Mobile reviews	Kaggle Amazon 200 reviews	English	KNN, MNB, BNB, SVM A: 95
[5]	ML techniques for Sentiment Analysis on product reviews	Kaggle Amazon 13057 reviews	English	NB: A: 98.17 SVM: A: 93.54
[6]	Sentiment Analysis for Mobile reviews	Online reviews	Kannada	NB A: 65 P: 62.5 R: 75 F: 68.2
[7]	Sentiment Analysis of Product Reviews	Amazon 400000 reviews	English	NB A: 66.95 SVM A: 81.77 DT A: 74.75
[8]	Sentiment Analysis of Mobile reviews	Kaggle 82815 reviews	English	NB A: 82.1 SVM A: 80.7 ME A: 82.7 DT

				A: 73.6
[9]	Sentiment Analysis of product reviews	JD.com 9600 reviews	Chinese	SenBERT-CNN A: 95.72 P: 95.21 R: 96.24 F: 95.32
[11]	Sentiment Analysis of mobile reviews	Online reviews 26988	English	Dictionary approach on mobile dataset A: 63 P: 65.3 R: 56.2 F: 60.3

2.3 Literature review for Sarcasm

Wen Z. et al. [13] introduced a neutral model by gathering auxiliary information. They used this information to find the context of the sentence. Bi-directional recurrent units based encoders were used to determine the presence of sarcasm in Chinese reviews.

Du Y. et al. [14] used a Convolution neural network to detect sarcasm by extracting emotions and semantics from the reviews. SenticNet library and Bi-directional LSTM was used to detect the features.

Sonawane S. et al. [15] used a co-occurrence matrix to detect the presence of sarcasm on a dataset from Twitter.

Dave A. et al. [16] worked on Hindi language reviews to detect the presence of sarcasm. Support Vector Machine classifier was used to detect sarcasm.

Sentamilselvan K. et al. [17] detected sarcasm & irony using Random Forest and Support Vector Machine techniques.

Afiyati R. et al. [18] took reviews of people who posted on Whatsapp in Indonesian language. For detecting sarcasm, they extracted sentiments, semantics and patterns from the reviews.

Katyayan P. et al. [19] used a rule based approach to detect sarcasm as well as hyperbole. Depending on the strength , the review was then categorized as being negative, positive or neutral.

Bouazizi M. et al. [20] extracted Twitter tweets and classified sarcasm into different categories depending on syntax, semantics, sentiment and patterns. Various machine language classifiers were used for the same with Random Forest giving the best performance.

Rao M. et al. [21] used machine language classifiers such as Random Forest, kNN and Support Vector machines to detect sarcasm.

Ashwitha A. et al. [22] used hash tags like #sarcastic and others using tweepee to collect tweets with emotions. POS tagging was done to filter the nouns, verbs, adjectives, etc. Different machine language classifiers were used to detect sarcasm.

Justo R. et al. [23] detected nastiness and sarcasm in the dataset. Naïve Bayes along with rule based techniques were used. They also emphasized on semantic and linguistic features as well as emotions expressed in the reviews.

Kumar H. et al. [24] used content based feature selection method to classify text being sarcastic or not. Initially, relevant features were extracted from the reviews. Clustering was carried out to group similar features in common clusters. SVM and Random Forest classifiers were used to determine sarcasm.

Mukherjee S. et al. [25] stressed on the fact that the linguistic style of authors was necessary to be known to detect sarcasm. They used Naïve Bayes classifier and fuzzy clustering algorithms.

Oprea S. et al. [26] used datasets of other researchers to detect sarcasm. Tag based supervision tweets gave a good accuracy score but tweets which were manually labeled had a comparatively less accuracy. They emphasized that sarcasm can be interpreted differently by different social and cultural groups.

Sundararajan K. et al. [27] proposed a probabilistic model with context and augmentation. Weights were calculated for every term pair. Word2vec model was used for word embeddings and Convolution Neural network technique was used to detect sarcasm.

Parameswaran P. et al. [28] used machine language classifiers to detect sarcasm in 3 datasets. Subsequently, LSTM method was used to detect the target of sarcasm expressed in the review.

Vinoth D. et al. [29] used particle swarm optimization algorithm in conjunction with Support Vector Machine classifier on a dataset from social media.

Barhoom A. et al. [30] used 21 classifiers amongst which majority were machine learning classifiers and a deep learning algorithm on two Kaggle datasets.

Muaad A. et al. [31] worked on two Arabic datasets to detect sarcasm & misogyny. BERT technique was used and gave an accuracy of 92% on binary classification and 82% on multiclass classification.

Li G. et al. [32] used graphs to extract dependency information and the model could catch complex forms of sarcasm. The model was trained on 6 different datasets.

Savini E. et al. [33] used pre-trained BERT model to detect sarcasm. Initially, emotion detection & sentiment classification was done. They worked on 3 datasets with each dataset differing in length and size.

Govindan V. et al. [34] used hash tags to find tweets related to COVID to find hyperbole in the tweets. Machine language classifiers were used with an accuracy score of 70%.

Omar A. et al. [35] worked on a Twitter dataset of Arabic language. They used machine learning and deep learning classifiers along with swarm optimization algorithm to find sarcasm. An accuracy score of 86.85% was achieved on the dataset.

Kumar R. et al. [36] used various machine learning classifiers and deep learning classifiers to detect sarcasm. GLOVE word embeddings were used to convert text to vectors. LSTM technique gave the best accuracy of 93.25%.

Razali S. et al. [37] manually handcrafted the feature sets. Fasttext technique was used to convert text to vectors. Logistic regression gave a f-score of 94%.

Kamal A. et al. [38] made a recurrent model to find self-deprecation sarcasm. Initially, features were extracted by passing word embeddings through a convolution layer. Attention layers were used with Adam optimizer and sigmoid function.

Abdelaal A. et al. [39] worked on Arabic text reviews using machine language classifiers. Decision tree classifier gave the best accuracy of 64.4%.

Kumar P. et al. [40] used FastText embedding technique along with the BERT transformer model to achieve an accuracy score of 98% on three publicly available datasets from Kaggle website.

Majumdar S. et al. [41] used rule based techniques to detect sarcasm using lexical analysis. Initially, opinion mining was done and depending on the sentence being sarcastic, the polarity was inverted.

Kumaran P. et al. [42] made a model to detect sarcasm, emotions and influential users on Twitter. Hash tags were detected and checked if there was positive sentiment in a pessimistic situation in the reviews. Bootstrap algorithm was used to detect sarcasm with an accuracy score of 87%.

Chen W. et al. [43] used GLOVE model for word embeddings and context incongruity to detect sarcasm. Two publicly available datasets were used and an accuracy of 78.28% was achieved.

Bharti K. et al. [44] used a Hadoop based framework for sarcasm detection from tweets. Hash tags related to sarcasm were used to extract the tweets. A corpus of words used universally was used to detect sarcasm.

Syrien A. et al. [45] extracted tweets of people on Bengaluru city traffic. TFIDF was used for word embeddings. Support Vector Machine gave the best accuracy of 80.98%.

Sharma K. et al. [46] used an ensemble model for detecting sarcasm. Word2Vec technique was used for word embeddings. They worked on two datasets and used LSTM technique with an accuracy score of 88.9%.

Kavitha K. et al. [47] used GLOVE technique for word embeddings. They combined CNN and RNN to detect the presence of sarcasm. Hyper parameter tuning was done at the end to increase the performance.

Goel P. et al. [48] worked with a model of a combination of CNN, bi-directional LSTM and GRU. Social media datasets were used but the model sometimes did not detect false negatives as well as sarcasm expressed politely.

Sharma K. et al. [49] used auto encoder techniques along with sentence based embeddings to detect sarcasm. BERT was used for the embeddings followed by LSTM for auto encoder. SoftMax function was used for final classification.

Parkar A. et al. [50] used machine language and deep learning classifiers to detect sarcasm on two publicly available datasets. Also, they proposed an ensemble model to detect sarcasm. BERT technique gave the best performance amongst the classifiers used.

Table 2.3 Summary of techniques to detect Sarcasm

References	**Objective of the paper**	**Dataset**	**Size**	**Language**	**Performance A:Accuracy P:Precision R: Recall F: F-score**
[13]	To detect Sarcasm using Sememe knowledge and auxiliary information	Guanchazhe Chinese Sarcasm Dataset	178,237 comments	Chinese	SAAG A: 72.07 P: 73.14 R: 73.05 F: 73.02 $BERT$ using $SSAS$ A: 76.00 P: 76.10 R: 75.91 F: 75.90
[15]	To detect sarcasm	Twitter	22000	English	TCSD method

	using co-occurrence		tweets		A: 94 P:94
[16]	To detect sarcasm using ML	Movie reviews	300 sentences	Hindi	Support Vector Machine A: 50
[17]	To detect Sarcasm using ML and rules	Online Dataset	Not mentioned	English	Accuracy Random Forest: 76 Support Vector Machine: 74
[18]	To detect Sarcasm on WhatsApp	Whatsapp	Not mentioned	Indonesian	Not mentioned
[19]	To detect Sarcasm using Sentiment strength	Social media	1500 sentences	English	Rule based extended algorithm A: 84 P: 68 R: 54 F: 59
[20]	To detect Sarcasm using patterns	Twitter	58609 tweets	English	Machine Learning algorithms Random Forest A: 83.1 P: 91.1 R: 73.4 F: 81.3 Support Vector Machine A: 60 P: 98.1 R: 20.4 F: 33.8 k nearest neighbor

					A: 81.5 P: 88.9 R: 72 F: 79.6 Maximum Entropy A: 77.4 P: 84.6 R: 67 F: 74.8
[21]	To detect sarcasm using ML	Amazon	Not mentioned	English	Accuracy: Random Forest: 62.34 k nearest neighbor: 61.08 Support Vector Machine: 67.58
[22]	To detect Sarcasm using ML	Twitter	Not mentioned	English	Accuracy: Support Vector Machine, Linear Support Vector Classifier, Random Forest, Decision Tree: 90 to 96
[23]	To detect Sarcasm and nastiness using ML	Online forums	617 posts	English	Naïve Bayes: A: 78.6 P: 77.4 R: 84.7

					F: 79.4
[24]	To detect Sarcasm using content based method	Amazon	1254 product reviews	English	1) Mutual Information + Clustering + Support Vector Machine A: 79.6 F: 75.2 2) Mutual Information + Clustering + Random Forest A: 77.2 F: 69.6
[25]	To detect Sarcasm using Clustering and Naïve Bayes	Twitter	2000 tweets	English	Fuzzy C means clustering + Content words + Naïve Bayes A: 65
[26]	To detect Sarcasm using context	Twitter	Riloff: 3200 tweets Ptacek: 50000 tweets	English	Neural models CASCADE Riloff: F: 47.8 Ptacek: F: 93.4 ED Riloff: F: 73.9 Ptacek: F: 88.7
[27]	Probabilistic model to detect Sarcasm	Twitter	40000 tweets	English	Accuracy: CNN with word2vec for contextual

					info: 85.67 CNN with word2vec for augmented model: 97.25
[28]	To detect Sarcasm using LSTM	Online reviews	224 tweets 506 books 950 reddit	English	Accuracy: TC-LSTM: 86(tweets) 71.5(reddit) 89(books) TD-LSTM: 85.1(tweets) 58.7(reddit) 89.1(books)
[29]	To detect Sarcasm using ML	Social media Kaggle	28501 sentences	English	Support Vector Machine classifier with Particle swarm optimization algorithm A: 94.7 P: 94.7 R: 95.2 F: 94.9
[30]	To detect Sarcasm using ML and DL	Kaggle news headlines dataset	28618 records	English	LSTM: A: 95.27 R: 96.62 P: 94.15 F-score: 95.37 Passive Aggressive

					classifier: A: 95.5 R: 96.09 P: 94.3 F-score: 95.19
[31]	To detect Sarcasm and Misogyny using Transformers	Twitter	15548 documents	Arabic	AraBERT classifier: Binary classification: A:91 Multiclass classification: A: 82
[32]	To detect Sarcasm using Graphs	IAC-V1 IAC-V2 Tweets-1 (Riloff) Tweets-2 (Ptácek) Reddit-1 (movies) Reddit-2 (technology)	IAC-V1: 1912 IAC-V2: 6520 Tweets-1 (Riloff): 1481 Tweets-2 (Ptácek): 53046 Reddit-1 (movies): 13910 Reddit-2 (technology):	English	Affection Enhanced Relational Graph Attention network IAC-V1: A: 72.2 P: 72.2 R: 72.2 F: 72.2 IAC-V2: A: 78.2 P: 78.42 R: 78.1 F: 78.2 Tweets-1 (Riloff): A: 85.8 P: 83.1 R: 76.2F: 79.7

			16015		Tweets-2 (Ptácek): A: 84.2 P: 84.2 R: 84.2 F: 84.2 Reddit-1 (movies): A: 75.8 P: 75.8 R: 75.8 F: 75.8 Reddit-2 (technology): A: 76.1 P: 76.1 R: 76.1 F: 76.1
[33]	To detect Sarcasm using Transformers	Internet Argument Corpus(Sarcasm V2) Reddit Twitter	4692 lines 1262434 comments 994 tweets	English	Sarcasm V2: BERT + TransferIMDB: F: 80.85 Reddit: BERT + TransferEmoNetSent: F:77.53 Twitter: BERT + TransferEmoNetSent: F: 97.43
[34]	To detect Sarcasm using ML	Chinese Twitter	6600 tweets	Chinese	Hyperbole Based Sarcasm Detection model:

					A: 75% P: 78% R: 63% F: 70%
[29]	To detect Sarcasm using ML	Kaggle	28501 posts	English	IMLB-SDC model: P: 94.7 R: 95.2 F: 94.9
[46]	To detect Sarcasm using Ensemble model	Social media, Headlines and Twitter	1956 tweets 26709 headlines	English	Twitter dataset A: 88.9 F: 81.5 Headlines dataset A: 81.4 F: 89.87
[47]	To detect Sarcasm using DL	News Headline Dataset Kaggle	26805 headlines	English	DLE SDC model A: 94.05 P: 94.06 R: 94.01 F: 94.03
[35]	To detect Sarcasm using Optimization algorithm	Semeval 2022 Twitter	3102 tweets	Arabic	ANN + Particle Swarm Optimization A: 86.85
[41]	To detect Sarcasm using Rules	Articles and writings	1500 articles	English	Algorithm and rule based with a database
[43]	To detect Sarcasm using context and clues	Reddit (Movies	Reddit mov:	English	Proposed model

		& technology) IAC (political debates)	8200 Reddit tech: 16094 IAC v1:1965 IAC v2:4646		(sentiment clues & incongruity) Reddit movies: A: 73.96 P: 73.98 R: 74.07 F: 73.42 Reddit technology: A: 74.53 P: 74.85 R: 74.45 F: 73.85 IAC v1: A: 68.53 P: 70.36 R: 68.77 F: 67.50 IAC v2: A: 78.28 P: 77.93 R: 77.58 F: 77.19
[36]	To detect Sarcasm using ML and DL	Kaggle news articles	Not mentioned	English	DT: A: 59.4 P: 85 R: 14 F: 72 KNN: A: 85.2 P: 97 R: 70 F:

					81 RF: A: 87.47 P: 95 R: 81 F: 88 SVM: A: 90.28 P: 94 R: 89 F: 92 CNN: A: 79.23 P: 83 R: 79 F: 74 LSTM: A: 93.25 P: 95 R: 90 F: 93
[37]	To detect Sarcasm using ML	News articles	32000 short news articles	English	SVM: P: 91 R: 90 F: 91 KNN: P: 86 R: 86 F: 86 LR: P: 95 R: 95 F: 94 DT: P: 90 R: 90 F: 90 DA: P: 91 R: 90 F: 91
[38]	To detect Sarcasm using DL	8 twitter datasets	D 1: 151283 D 2: 3892 D 3: 1801 D 4:	English	CAT-BiGRU model A: 93 P: 92 R: 98 F: 94

			15060 D 5: 52576 D 6: 41703 D 7: 42622		
[42]	To detect Sarcasm using mathematical modeling	Twitter	1.5 million tweets	English	PID-EDSDISI method: A: 87 P: 83 R: 80 F: 82
[39]	To detect Sarcasm using ML	Arabic texts	10000 tweets	Arabic	Decision Tree: 64.4
[49]	To detect Sarcasm using Encoders	Reddit, headlines, tweets	1956 tweets 26709 headlines	English	Accuracy: Reddit: 83.92 Headlines: 90.8 Twitter: 92.8
[14]	To detect Sarcasm using DL	Twitter and Reddit datasets	13479 tweets 31822 comments	English	Proposed method Bi-LSTM with SenticNet Twitter: A: 73 F: 76 SARC: A: 71 F: 71
[45]	To detect Sarcasm using ML	Twitter	20500 tweets Bengaluru traffic	English	Accuracy Naïve Bayes: 59.97 Logistic

					Regression: 79.93 Support Vector Machine: 80.98 Random Forest: 77.38
[44]	To detect Sarcasm using MapReduce	Twitter	1.45 million tweets	English	MapReduce function with Hadoop framework and corpus: F-score: 97
[48]	To detect Sarcasm using DL	News Headlines, Reddit	44263 headlines 899955 comments	English	Ensemble model News headlines A: 99 Reddit A: 82
[40]	To detect Sarcasm using Transformers	News Headlines, Reddit, Twitter	26709 headlines 1 million reddit 39780 tweets	English	FastText + BERT A: 98.25 P: 92 R: 98 F: 98.32
[51]	Using DL and ML techniques to detect sarcasm	SARC & Headlines	Reddit News	E	SARC LR A: 70.5 F: 69.3 Headlines Encoder

					A: 80.1 F: 75.5
[52]	To detect sarcasm for Alexithymia Individuals	Twitter corpus	137032 tweets	E	BERT A: 83
[53]	Detecting sarcasm for Indonesian reviews	Reddit Twitter	14116 comments	Indonesian	Reddit F: 62.7 Twitter F: 76.9
[54]	Using Chicken Swarm Optimization & Graph Neural Networks to find sarcasm	Kaggle	1956 samples	E	CSOA-GNN A: 92.8
[55]	DL techniques to detect sarcasm	News headlines	Not mentioned	E	BERT A: 88
[56]	ML and DL models to detect sarcasm	News headline	28619 records	E	RNN A: 79 F: 76
[57]	Sarcasm Detection in Libyan Arabic Dialects	Facebook	5082 dialects	Arabic	SVM A: 79.15 P: 79.3 R: 79.7 F: 79.5
[58]	Using LLM's to detect sarcasm	Reddit Headlines Chinese Internet reviews	30000 reviews 28619 reviews 2000 texts	E Chinese	BERT FT Headlines: F: 90 Reddit: F: 71 Chinese I: F: 94

[59]	Using transformers to detect sarcasm in Arabic	Twitter	10547 tweets	Arabic	Ara-BERT A: 87.8
[60]	ML techniques to detect sarcasm	Facebook	5471 comments	E	Hierarchical clustering Silhouette score: 0.032
[61]	Detect sarcasm in Urdu tweets	Twitter	12910 tweets	Urdu	BERT A: 79.51 F: 80.04
[62]	Using transformers to detect sarcasm	Corpus V2 Reddit Headlines	3260 posts 10000 reviews 14000 reviews	E	BERT with fuzzy logic F: 89.54
[63]	Using transformers to detect sarcasm	Twitter	19816 images	E	BERT F: 95.79
[64]	ML techniques to detect sarcasm	News headlines	20000 reviews	E	LSTM A: 88 F: 88
[65]	Sarcasm Detection using DL techniques	News headlines	28619 reviews	E	BiLSTM A: 86
[66]	Sarcasm detection using DL techniques	Reddit	1118984 0 reviews	E	MHA-BiLSTM F: 79.64
[67]	Detecting sarcasm in multiple modalities	MUStARD dataset	6365 videos	E	Model F: 75.2
[68]	Ensemble model to detect sarcasm	News headlines	28619 reviews	E	Ensemble model

					A: 93
[69]	Using DL techniques to detect sarcasm in Chinese	Online product reviews	42830 reviews	Chinese	GRU-CAP F: 96.38
[70]	Detect emojis for sarcasm detection	Online reviews	29377 texts	E	DeepMoji model F: 68.14
[71]	Transformers to detect sarcasm in Arabic	Twitter	3000 tweets	Arabic	AraBERT model F: 87
[72]	ML technique to find emoticons in sarcasm detection	Twitter	15657 tweets	E	Decision Tree + Library F: 80
[73]	Transformer model to detect sarcasm in Arabic	Twitter	26014 tweets	Arabic	Transformer model F: 68.28
[50]	Ensemble model to detect sarcasm	News headlines Reddit	28619 reviews 80000 reviews	E	Ensemble model F: 81
[74]	Fuzzy neural networks to detect sarcasm	Mustard Memotion datasets	6992 reviews	E	QFNN model
[75]	A dual channel mechanism to detect sarcasm	IAC Twitter	5216 rev 3526 rev	E	BNS-Net F: 79.23
[76]	Multimodal sarcasm detection	Online dataset	19816 records	E	MuMu network

					F: 90.40
[77]	Transformers for sarcasm detection in Indonesian	X	8700 tweets	Indonesian	IndoBERT F: 95
[78]	Multimodal sarcasm detection	Online dataset	19816 records	E	FSICN RoBERTa F: 93.45
[79]	Transformers to detect sarcasm using machine translation	SemEval	6520 reviews	E Chinese	m-BERT 50 model F: 79.5

2.4 Literature review for detecting Implicit Aspects

Schouten K. et al. [80] used co-occurrence technique to detect implicit aspects from mobile phones dataset and restaurant dataset. They counted the frequency of co-occurrence and the highest score was labeled as the missing implicit aspect.

Eldin S. et al. [81] used cuckoo search algorithm to detect implicit aspects in English and Arabic datasets. Their approach gave a better performance compared to rule mining and conditional random field techniques. The highest weighted score was assigned as the implicit aspect.

Omurca S. et al. [82] used Laplace smoothing method and weights were calculated between features and sentiment words. Naïve Bayes method was used with a F-score of 77%.

Liu Z. et al. [83] used fuzzy c means algorithm to split the aspects in different classes. Association rules were made to find the implicit aspects. They worked on Chinese reviews of an ecommerce website.

Ekinci E. et al. [84] used semantic similarity based topic model. Dataset was of restaurant reviews and Babelfy was used to extract aspects. Topic assignment was done using LDA and implicit features were found out.

Zeng L. et al. [85] extracted Chinese reviews of mobiles and cameras from Amazon. Initially, rule based method was used to extract all explicit feature opinion pairs. Clustering was done to group similar objects together. A centroid based classification method was used to identify the implicit aspects.

Abdullah R. et al. [86] framed grammatical rules to extract explicit aspects and a hybrid approach to gather implicit aspects. SentiCircle technique was used to extract sentences with explicit and implicit aspects. Co-occurrence technique was used to find the implicit aspects. Dataset was of restaurant reviews from Trip Advisor website.

Hajar H. et al. [87] used a dictionary based approach to detect implicit features. Naïve Bayes classifier was used with a hybrid model to increase the performance metrics of the model.

Afzaal M. et al. [88] took reviews of hotels and restaurants to gather implicit as well as explicit aspects. Fuzzy algorithms were used which gave a better performance than traditional machine learning classifiers.

Bhatnagar V. et al. [89] extracted implicit and explicit aspects using Conditional Random Fields technique. SentiWordNet resource along with frequently occurring nouns was used to gather the explicit aspects. Implicit aspect recognizer was used to gather the implicit aspects. Dataset was on the tourism domain.

Wan Y. et al. [90] gathered implicit as well as explicit aspects using dictionary approach and POS tagging. The dataset was of an ecommerce website. The objective was to improve the performance of opinion mining by considering the implicit as well as explicit aspects.

Sun L. et al. [91] found out the similarity between the opinion words and product features. Co-occurrence matrix was used to find the relation between the opinions and the features. Context was also considered while finding the implicit aspects.

Nandhini M. et al. [92] used co-occurrence matrix for opinion words and features. If the review did not have an explicit aspect, then they used a ranking method based on frequency to map the opinion words with the required implicit feature.

Rana A. et al. [93] gathered clues from reviews where explicit aspect was missing. Normalized Google distance was then used to find the implicit aspect.

Rana A. et al. [94] used a hybrid approach to extract implicit aspects based on co-occurrence of opinions and aspects as well as Normalized Google Distance. In every sentence, explicit aspects were searched for. If none, were found the list of explicit

features was checked and the most likely explicit feature was categorized as the implicit feature.

Wang W. et al. [95] used collocation extraction techniques to find co-occurrences between implicit and explicit features. Hybrid rules were framed using the co-occurrences. A semi-supervised model using dependency grammar was used to find the implicit aspects.

Sayali A. et al. [96] took reviews of tourism domain and used point wise mutual information to gather implicit and explicit aspects. SentiWordNet was used to determine the polarity.

Xu H. et al. [97] extracted implicit features using Support Vector Machine and Latent Dirichlet Allocation techniques. They made an improvement over the LDA model using context information. Clusters were made and features were extracted from the dataset. SVM was trained and used to gather the implicit aspects.

Sun L. et al. [98] used topic modeling to gather implicit aspects. Opinions were extracted and split into general opinions and special ones. The assumption was that general opinions could contain many aspects while special opinions contained exactly one aspect. Probability distributions were used to find the implicit aspects.

Liu L. et al. [99] used clustering to find the aspects from product reviews. Opinions were categorized into clear and vague opinions. In the clear opinions, there could be an implicit aspect while in the vague opinions, context information was needed. Clusters were made and every cluster had words which were similar to one another. Reviews which had a missing explicit aspect and were vague opinions were replaced by an entity. If the opinion was clear, then it was mapped to one of the existing clusters.

Yan Z. et al. [100] used an extended PageRank algorithm to find aspects. Pairs of features and opinions were extracted. Node Rank algorithm was used to find associations between the explicit aspects and their respective opinions. In case an opinion was present with no

explicit aspect then the aspect with the highest Node Rank value was categorized as the implicit aspect.

Selamat A. et al. [101] gathered aspects from tweets using a combination of association rule mining, SentiWordNet and dependency parsing. The dataset was on hate crimes.

Lazhar F. et al. [102] extracted implicit aspects using Ontology technique and dependency grammar to finf the pairs of opinions and explicit aspects. Hotel reviews from Trip Advisor website were used and for classification Support Vector Machine was used.

Marstawi A. et al. [103] identified features using Ontology technique. They suggested rather than using a lexicon approach to detect implicit aspects, the technique of Ontology gave a better performance.

Ray P. et al. [104] used a rule based method along with CNN to detect features from restaurants and laptop reviews. To gather the aspects, POS tagging was done. Initially, CNN technique was used followed by the rule based method.

Omar M. et al. [105] proposed a Bi-LSTM model using domain trained word embeddings. They worked on extracting multiple implicit aspects with a F-score of 83%.

Setiowati Y. et al. [106] split the task of finding implicit aspects into two stages. In the learning stage, co-occurrence matrix was used to extract opinions and their categories. In the validation stage, work was done to find opinion words that matched the implicit opinion sentences.

Cao H. et al. [107] made a model which could capture features along with their categories and sentiments. BERT model was used and the datasets were of laptop and restaurant reviews.

Thuy L. et al. [108] used a hybrid approach with context vectors to detect aspects and classify them for sentiment. Pairs of sentiment with their aspects was made and passed to a deep learning classifier. Ontology and dependency grammar was used in conjunction.

Dosoula N. et al. [109] used a score function with Logistic regression classifier. They worked on finding multiple implicit aspects in sentences. They used an annotated dataset of restaurant reviews with one implicit feature per review.

Tubishat M. et al. [110] worked on finding aspects in reviews. Explicit aspects were extracted using whale optimization algorithm. A hybrid model was suggested using a combination of corpus, co-occurrence, dictionary and web based similarity to detect implicit aspects.

William et al. [111] extracted 5000 sentences of hotel reviews of Indonesian language. They used a pre trained transformer model to find out the aspects along with the opinions.

Li Y. et al. [112] used BERT model on a dataset of 1150 clothing reviews of Chinese language. They manually annotated reviews with the assumption that there is only one implicit aspect per sentence.

Verma K. et al. [113] took reviews from airline industry and used machine language classifiers to detect implicit aspects. Word2Vec technique was used for word embeddings. Manual annotation was done for the implicit aspects. XGBoost technique gave the best F-score amongst the classifiers used.

Omar M. et al. [114] proposed a novel unsupervised probabilistic model based on Latent Dirichlet Allocation to check for semantic regularities and to extract explicit aspects. The dataset was of restaurant reviews.

Mar A. et al. [115] used clustering technique to distinguish between two dissimilar meaning words. Reviews were further categorized into implicit and explicit types. Mapping was done to see if there were implicit features. BERT was used for classification and detecting implicit aspects.

Halima B. et al. [116] used WordNet dictionary in conjunction with KNN classifier to find implicit features. It proved an improvement over the traditional KNN classifier with a F-score of 77.6% and eliminating the problem of overfiitting and underfitting.

Parkar A. et al. [117] proposed a combination of clustering, co-occurrence matrix and classification approach to detect implicit aspects.

Table 2.4 Related work for Implicit Aspects

References	**Objective**	**Dataset**	**Methodology**	**Performance Metrics**
[80]	detecting the right implicit feature	Mobile phone and restaurant	10 fold cross validation, POS filter	F-score: Mobile phone: 12.9 Restaurant: 63.3
[81]	detecting implicit features and explicit features	English and Arabic datasets	Cuckoo search	Recall: Feature type: Specific: 88 General: 54 Context: 60 Feature extraction: 87
[82]	to detect implicit aspects	Turkish reviews	Laplace smoothing method, Naïve Bayes	F-measure: 77

[83]	to detect implicit aspects using association rules	Chinese reviews	Fuzzy C means, Simulated Annealing	Precision: 83.26 Recall: 68.37 F-score 75.08
[84]	to detect implicit aspects using Latent Dirichlet Allocation	Restaurant reviews	Latent Dirichlet Allocation, Babelfy	Not mentioned
[85]	to detect implicit features using centroid classifier	Chinese ecommerce reviews	Centroid classifier (mathematical formulae)	Precision 85.59 Recall 72.93 F-score: 78.76
[86]	to extract aspects in different sentence structures	Trip Advisor Reviews	SentiCircle, grammatical rule, hybrid approach	F-score: SentiCircle: 84 SC + Opinion

				Lexicon: 86 SC + OL + Implicit Aspect Lexicon + Co-occurrence: 89

Table 2.5 Supervised techniques to detect Implicit Aspects

References	**Objective of the paper**	**Dataset**	**Size**	**Language**	**Performance A:Accuracy P:Precision R: Recall F: F-score**
[85]	To detect implicit features using Rules	Cell phones and Digital cameras from Amazon (China)	4083 reviews 12760 sentences	Chinese	Rule based method for explicit feature opinion pairs: Dataset of cell phones P: 80.21 R: 79.99 F: 80.1 Dataset of digital cameras P: 81.95 R: 83.43 F: 82.68 Implicit feature identification using centroid method: Dataset of cell phones

					P: 82.07 R: 68.48 F: 74.66 Dataset of digital cameras P: 85.59 R: 72.93 F: 78.76
[87]	To extract adjectives to detect implicit aspects	Electronic products (Apex, canon, jukebox, Nikon and nokia) and restaurant reviews	Electronic products: 314 reviews Apex: 99 rev Canon: 45 rev Jukebox: 95 rev Nikon: 34 rev Nokia: 41 rev Restaurant reviews: 3044 sentences	Indian	Dataset of electronic products Apex: P: 93 R: 94 Canon: P: 94.3 R: 91.4 Jukebox: P: 97.1 R: 86 Nikon: P: 96 R: 92 Nokia: P: 97.6 R: 88.8 Restaurant: P: 97.6 R: 94.3

[88]	To mine tourist reviews for aspects using fuzzy logic	Trip Advisor, Open Table websites for restaurants and hotels	Restaurant: 2000 reviews Hotel: 4000 reviews	English	Restaurants Dataset: FURIA: A: 90.12 P: 89 R: 90 F: 87 Hotels Dataset: FLR: A: 86.02 P: 85 R: 86 F: 85
[89]	To detect implicit aspects using Supervised approach	Tourism (Trip Advisor)	Not mentioned	English	Not mentioned
[91]	To detect Implicit Aspects using context	datatang.com, 360buy.com	Mobile phone: 3656 reviews Computer: 2446 reviews	Chinese	Computer reviews: P: 64 R: 67 Mobile phone reviews: P: 70 R: 74
[92]	To detect implicit aspects using co-occurrence and ranking	Online reviews	Not mentioned	Not mentioned	Not mentioned

Table 2.6 Semi Supervised Techniques to detect Implicit Aspects

References	**Objective of the paper**	**Dataset**	**Size**	**Language**	**Performance A:Accuracy P:Precision R: Recall F: F-score**
[93]	To detect implicit aspects using knowledge based approach	Product reviews: Electronic products	Electronic products: 3945 sentences	English	P: 77 R: 90 F: 83
[94]	Using hybrid approach to detect aspects	Online Customer reviews	Reviews: 314 Sentences: 4259	English	Not mentioned
[95]	To detect aspects using association mining	Product Reviews	Sentences: 14218	Chinese	P: 86 R: 67 F: 75.51
[96]	To detect aspects using sentiment	Tourism: Trip Advisor	Reviews: 300	English	P:72 R: 66
[97]	Using mining to detect implicit features	Product reviews	Sentences: 10183	Chinese	P: 87.42 R: 70.05 F: 77.78

[98]	To extract implicit features using topic modeling	Product Reviews	Reviews: 5731	Chinese	P: 76 R: 66.5

Table 2.7 Unsupervised Techniques to detect Implicit Aspects

References	**Objective of the paper**	**Dataset**	**Size**	**Language**	**Performance A:Accuracy P:Precision R: Recall F: F-score**
[99]	To extract features using mining	Product reviews	Reviews: 3500 Sentences : 9998	Chinese	P: 72.5 R: 64.5
[100]	To extract aspects using page ranking method	Product reviews	Reviews: 29192	Chinese	P: 80.5 R: 70 F: 73.9
[101]	To gather aspects using sentiments	Hate crime reviews	Tweets: 622	English	Not mentioned

[102]	To gather aspects from hotel reviews	Hotel reviews: Trip Advisor	Reviews: 150	English	P: 93 R: 87 F: 90

Table 2.8 Techniques to detect Implicit Aspects

References	**Objective of the paper**	**Dataset**	**Size**	**Language**	**Performance A:Accuracy P:Precision R: Recall F: F-score**
[104]	To detect aspects using deep learning and rule based method	SemEval	Laptop rev: 3045 Restaurant: 3041	English	CNN + rule based method A: 87
[105]	To detect implicit aspects using Bi-LSTM	SemEval	Restaurants (2 datasets) 3841 + 2086 reviews	English	Bi-LSTM using Domain trained word embeddings P: 84 R: 81 F: 83
[106]	To extract implicit opinions using Co-occurrence	Hotel user reviews, TripAdvisor	3055 reviews	Indonesian	Co-occurrence matrix: A: 83.87
[107]	To detect aspects with sentiment using transformers	SemEval dataset Laptop rev: 4076 Restaurant: 2286	Laptop: 4076 Restaurant: 2286	English	Laptop: P: 41 R: 37 F: 39 Restaurant: P: 56 R: 51 F: 53
[108]	To identify	Corpus 12	Mobile:	English	ML-KB

	implicit aspects using deep learning	Mobile reviews	69,905 samples		P: 71 R: 83 F: 76
[109]	To detect multiple implicit features	Restaurant	3044 sentences	English	F: 64.5
[110]	To detect aspects using optimization and hybrid approach	Not mentioned	Not mentioned	Not mentioned	Not mentioned
[111]	To extract opinion triplet using pretrained model	Hotel Reviews	5000 sentences	Indonesia	F: 79.5
[112]	To detect implicit aspects using BERT	Clothing Reviews	1150 implicit aspect sentences	Chinese	F: 96
[113]	To detect implicit aspects	Airlines	Few reviews	English	ML algorithms F: 94.7
[114]	To detect implicit aspects	Restaurant	587095 reviews	English	Topic Seeds LDA P: 89
[115]	To detect	Mobile	3020	English	BERT

	implicit aspects using supervised learning	reviews	reviews		A: 82
[116]	To detect implicit aspects using dictionary and supervised approach	Electronic products Restaurant reviews	3044 reviews	English	KNN + WordNet F: 77.6
[80]	To find implicit aspects	Restaurant and product reviews	Not mentioned	English	Product F: 12.9 Restaurant F: 63.3
[81]	To find implicit aspects using optimization technique	Product reviews	Not mentioned	Arabic and English	Cuckoo search optimization algorithm P: 87
[82]	To extract implicit aspects using graph based method	Hotel reviews	1000	Turkish	Graph Laplace smoothing F: 77
[83]	To extract implicit aspects using association	Mobile reviews	2265	Chinese	Fuzzy C means + Association rules F: 75.08

	rules				
[84]	To extract implicit aspects using LDA	Restaurant reviews	2647	English	LDA Not mentioned
[40]	To detect implicit aspects using rule based and clustering	Mobile and camera	4083	Chinese	Rule based + Clustering F: 78.6
[86]	To detect implicit aspects using rules, dictionary and co-occurrence	Restaurant	Not mentioned	English	SentiCircle+Co-occurrence+ Lexicon F: 89
[93]	To detect implicit aspects using co-occurrence	Products	3945	English	Co-occurrence + Normalized Google distance F: 83
[87]	To detect implicit aspects by using hybrid approach	Electronic products and Restaurant	E: 314 R: 3044	Chinese	Electronics P: 97 Restaurant P: 97
[90]	To detect	Clothes	7300	Chinese	Dictionary + rule based

	implicit aspects using rules and dictionary				
[88]	To detect aspects using Fuzzy logic	Restaurants Hotels	R: 2000 H: 4000	English	Fuzzy logic A: 90
[89]	To detect implicit aspects using Supervised approach	Tourism	Not mentioned	English	Conditional Random Fields
[91]	To detect implicit aspects using co-occurrence	Computer Mobile phone	C: 2446 M: 3656	Chinese	Co-occurrence Computer: P: 64 R: 67 Mobile: P: 70 R: 74
[94]	To detect aspects using rules	E-commerce products	Not mentioned	English	Rule based
[92]	To detect implicit aspects using co-occurrence and ranking	Online reviews	Not mentioned	English	Co-occurrence + Ranking
[95]	To detect implicit	Mobile reviews	1816 reviews	Chinese	Co-occurrence + Rule P: 86 R: 67 F: 75

	features using co-occurrence and rules				
[117]	To detect implicit features using co-occurrence and clustering	Mobile reviews	Not mentioned	English	Co-occurrence + Clustering + Classification
[96]	To detect implicit sentiments using co-occurrence and association rules	Tourism	Not mentioned	English	Co-occurrence + Association Rules P:0.72 R: 0.66
[97]	To detect implicit features using topic mining	Mobile reviews	14218 sentences	Chinese	LDA + rules P: 87 R: 70 F: 77
[98]	To detect implicit aspects using topic opinion model	Mobile comments	3420 comments	Chinese	LDA + Extension P: 76 R: 66

[99]	To detect features using Clustering	Computer + Mobile	3500 reviews	Chinese	Clustering P: 72 R: 64
[100]	To detect features using page rank algorithm	Camera Mobile DVD player reviews	C: 8901 M: 11291 D: 9000	Chinese	PageRank algorithm P: 80 R: 70 F: 73
[101]	To detect aspects using dependency parsing and rules	Twitter Hate crime reviews	622 tweets	English	Dependency Parsing + Rule A: 75 P: 84 R: 65
[102]	To detect implicit opinions using Ontology and grammar	Hotel reviews	150 reviews	English	Ontology + dependency grammar P: 93 R: 87 F: 90
[103]	To detect aspects using Ontology	Product reviews	Not mentioned	English	Ontology P: 87 R: 88 F: 87

Chapter 3

Proposed Framework

3.1 Proposed Framework for Sarcasm Detection

In this thesis, a hybrid framework is proposed to detect sarcasm called as CB framework. The deep learning technique of Convolution Neural Networks and Bi-directional Encoders Representations from Transformers are combined to detect sarcasm.

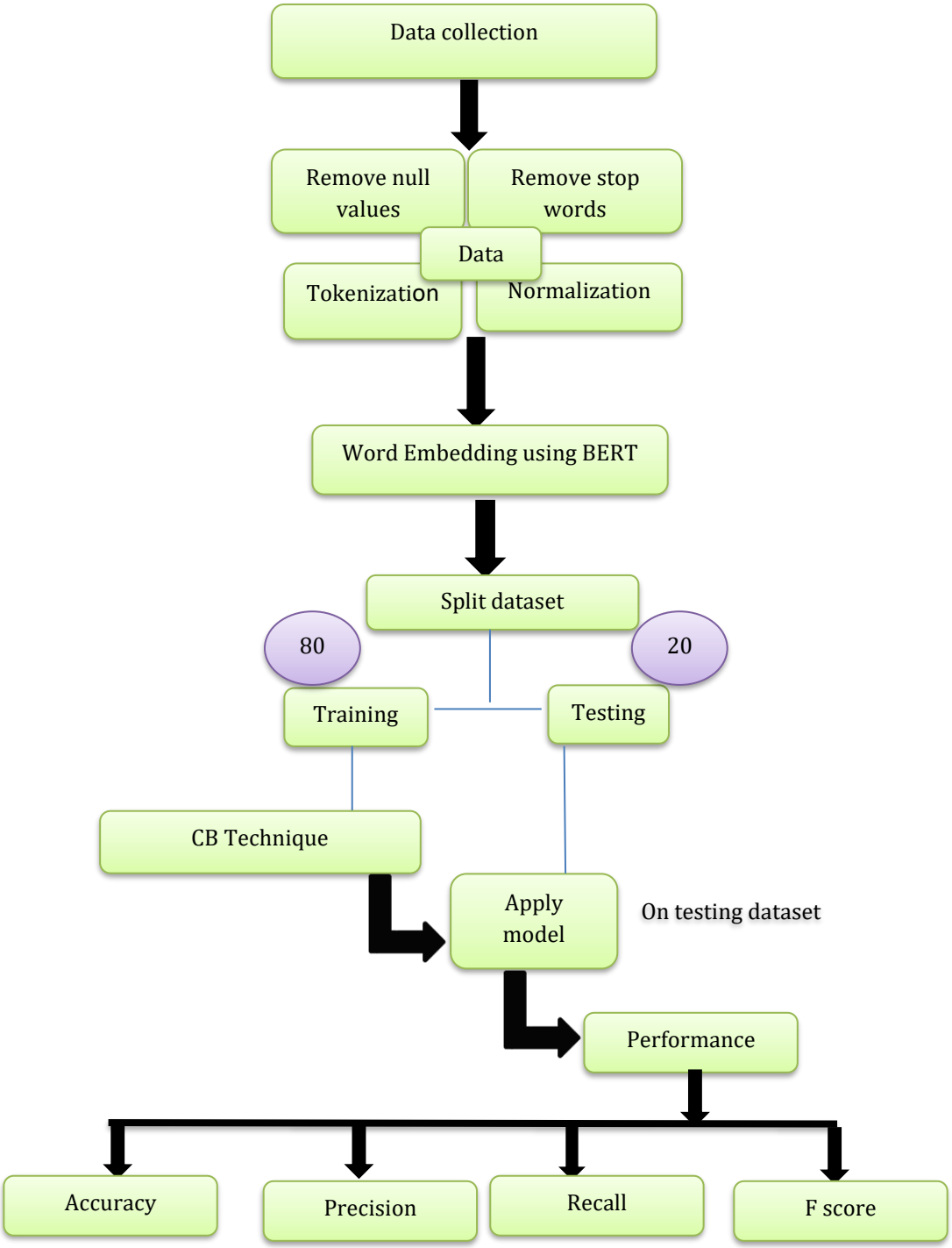

Figure 3.1 Proposed framework to detect Sarcasm

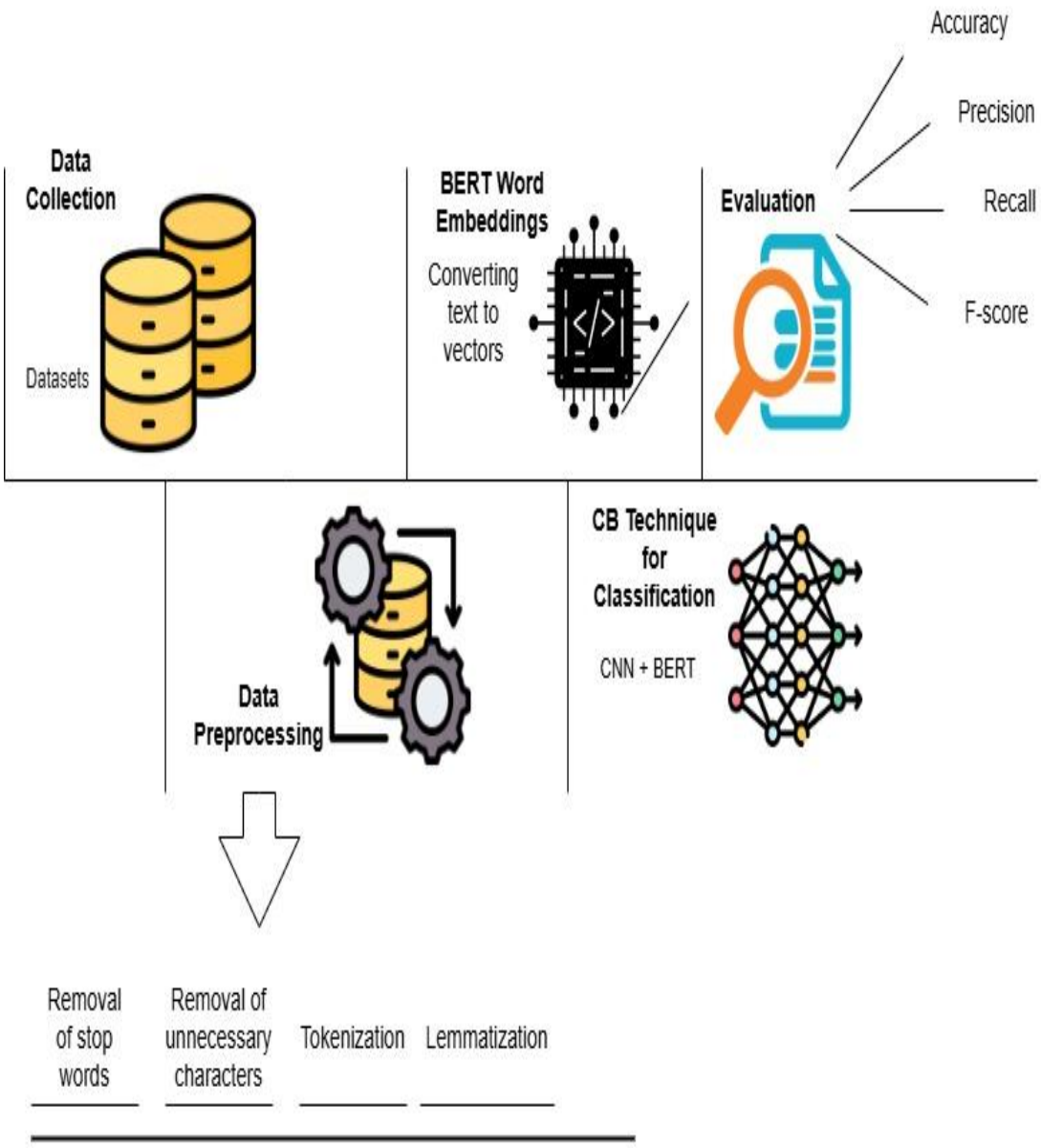

Figure 3.2 Architecture Diagram for Sarcasm Detection

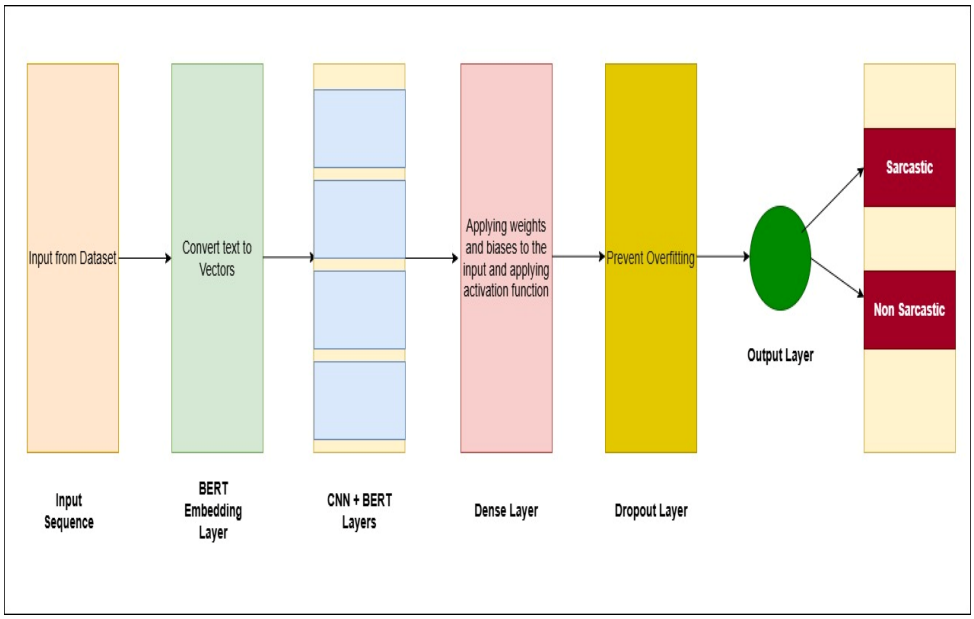

Figure 3.3 Internal Architecture for Sarcasm Detection

In this thesis, a hybrid framework is proposed to detect sarcasm called as CB framework. The deep learning technique of Convolution Neural Networks and Bi-directional Encoders Representations from Transformers are combined to detect sarcasm.

3.2 Proposed Framework for Detection of Implicit Aspects

In this thesis, a novel framework has been proposed for detecting implicit aspects from text reviews called as EDS framework. The encoder decoder technique of transformers along with a backup of supervised learning has been proposed in this study.

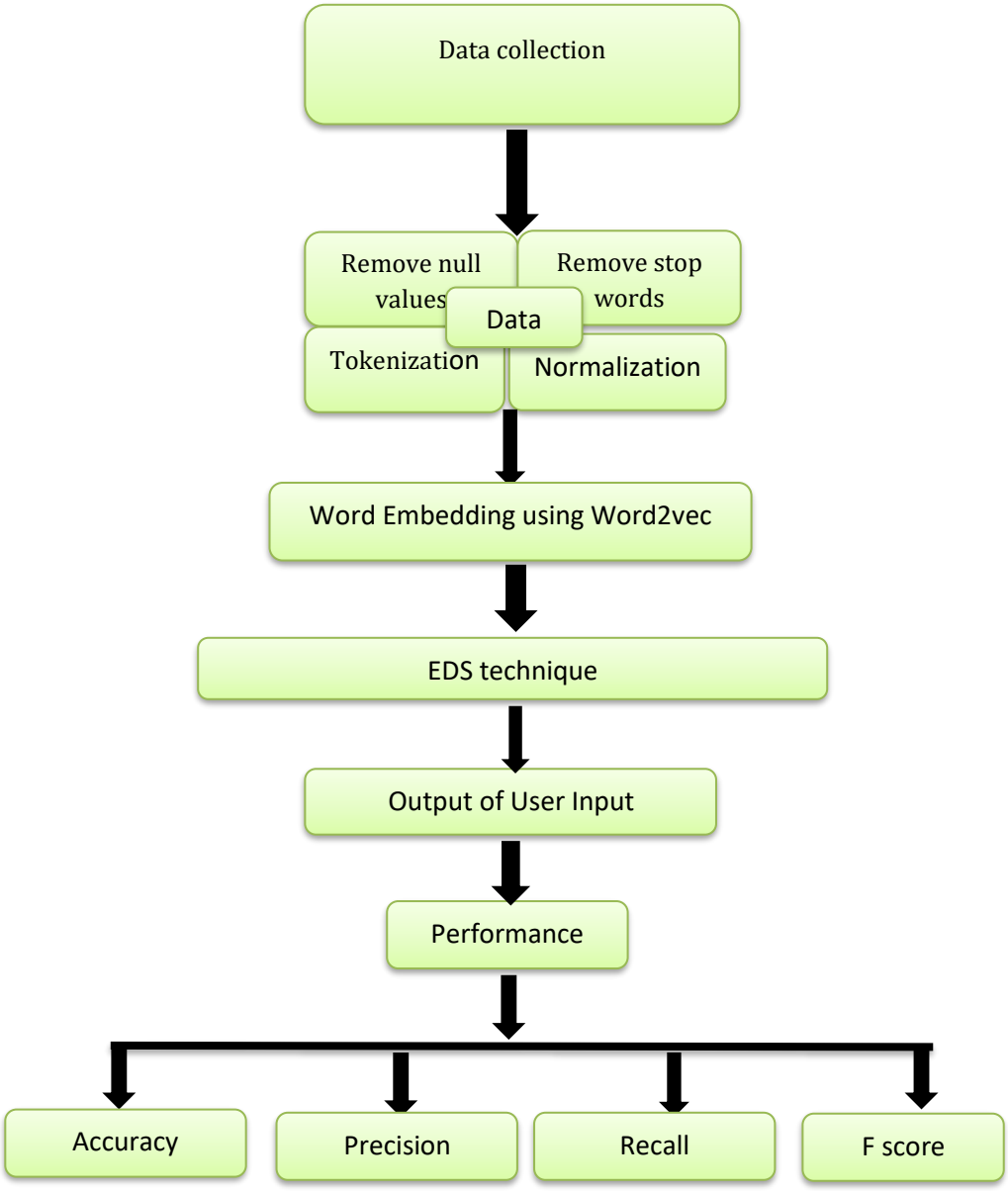

Figure 3.4 Proposed framework to detect Implicit Aspects

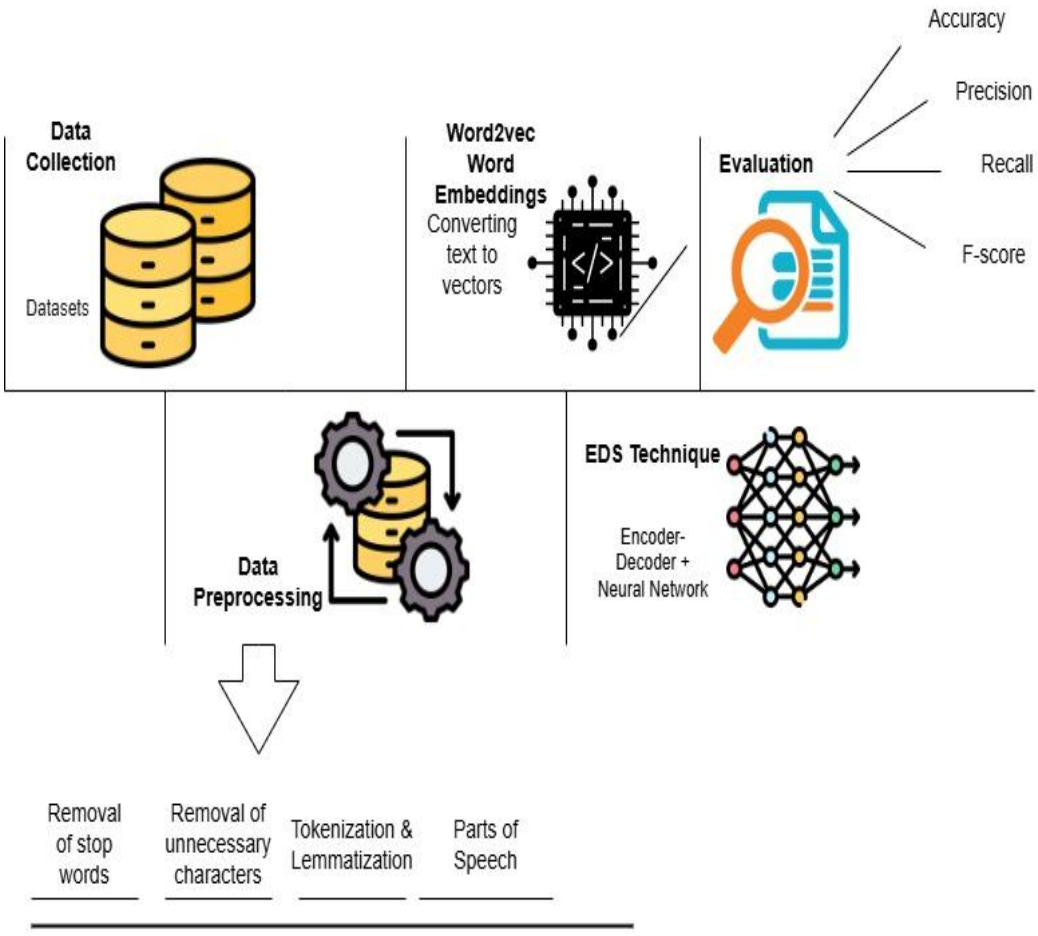

Figure 3.5 Architecture Diagram for Implicit Aspect Detection

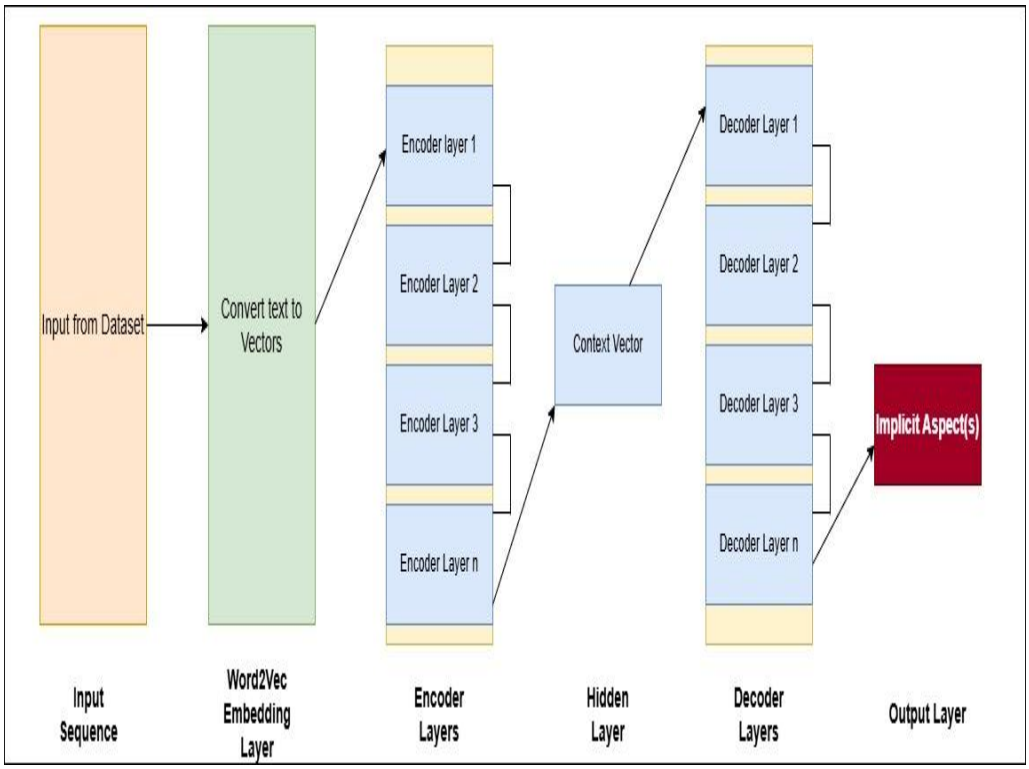

Figure 3.6 Internal Architecture for Implicit Aspect Identification

Detailed Methodology for the task of Sarcasm Detection and Implicit Aspect Detection is explained in Chapter 5.

Chapter 4

Implementation of CB Framework for Sarcasm Detection

4.1 Techniques to detect Sarcasm in text reviews

Initial research on detection of sarcasm was focused on syntactic features, semantics, pattern matching, etc. Research nowadays is focused on machine learning and deep learning techniques. Few of the techniques are discussed here:

4.1.1 Lexical Analysis

Lexical analysis is one of the basic steps in the domain of Natural language processing. A lexical analyzer is required to do lexical analysis. It scans the entire sentence/review and groups words into different individual tokens. Some examples of tokens are punctuation marks, constants, operators, etc. These tokens are required in the pre-processing steps of sentiment analysis. By doing lexical analysis, we are cleaning the data, simplify the input and make the original dataset smaller by removing unwanted symbols.

Majumdar S. et al. [41] checked for sentiment of sentences initially by using rule based methods & taking random input of users for the dataset. Lexical analysis was done using a rule based approach to detect sarcasm & by taking a pre-trained categorized database. If sarcasm was found then the sentiment of the sentence/review was inverted. At the end, sentiment analysis was done on the entire dataset to improve the performance metrics.

Kumaran P. et al. [42] made a model to detect sarcasm by looking for hash tags in tweets from the X website. Parts of Speech tagging was done initially and then bootstrap algorithm was used for detecting sarcasm. Lexical analysis was done initially after extracting the tweets.

Afiyati R. et al. [18] used lexical analysis by matching the patterns in data extracted from Whatsapp in Indonesian language. Syntactic related features, semantic features, sentiment features, pattern matching features, etc. were used.

4.1.2 Word Embedding

Word embedding is used in Natural language processing to convert text to vectors. Word embedding can be done using frequency or prediction based technique. Fast Text, Word2vec, Bag of words, Term frequency Inverse document frequency, GLOVE, BERT, etc. are used for word embedding.

Chen W. et al. [43] worked on two linguistic features and mentioned that sentiment and incongruity information is beneficial to detect sarcasm. They used soft sentiment labels and semantic information was used. Two publicly available datasets of Reddit and IAC were used. GLOVE was used for word embedding.

Oprea S. et al. [26] worked on two datasets to detect sarcasm. The dataset was from X website with one dataset being manually labeled and the other was tag based labeled. Historical data of X users was considered while making the word embedding.

4.1.3 Context

In context based sarcasm detection, the knowledge of the person giving the review is taken into consideration. The way the person has written reviews in the past is also a way to judge the context of the review and chances of it being sarcastic or not. Some models judge the context by grouping of words or sequences. Relationships if any are found related to the past data.

Katyayan P. et al. [19] used ELMo sentence embeddings for retaining context information. They used baseline models such as GRU, LSTM, RNN, Bi-

directional LSTM to detect sarcasm. Reddit corpus was used and an analysis of different techniques resulted in GRU model to be the best amongst the ones used.

Sundararajan K. et al. [27] made a probabilistic model to extract the word pairs in the reviews and they utilized contextual information while doing it. The weights of the word pairs were dependent on the context information. Word2vec model was used for word embeddings and CNN technique was used to detect sarcasm.

Bharti S. et al. [44] extracted tweets from website X using hash tags related to sarcasm. A corpus of universal words was used and depending on the presence of the words in the tweets, the presence of sarcasm was checked.

4.1.4 Machine Learning

Machine learning is a subset of Artificial Intelligence that has different techniques and models to make computers understand and learn without help of humans. Large amounts of data are needed for machine learning techniques to analyze data and find patterns in the data.

Kumar K. et al. [24] proposed a novel technique to find sarcasm in Amazon product reviews. In the initial stage, relevant aspects were gathered by using Chi-Square, Information gain and mutual information. Clustering was done to group similar aspects. Random forest and Support Vector machine techniques were used to find sarcasm.

Barhoom A. et al. [30] used 21 machine learning and 1 deep learning algorithm to detect sarcasm in headlines news dataset available on Kaggle website.

Syrien A. et al. [45] used machine language classifiers such as Logistic Regression, Linear Support Vector, Random Forest & Gaussian Naïve Bayes to detect sarcasm on tweets extracted from website X.

Ashwitha A. et al. [22] worked on understanding the context and emotion behind the reviews. Sarcastic words based on context were selected and then it was judged if a review is sarcastic or not. Tweets were extracted and used as reviews.

Rao M. et al. [21] detected sarcasm in Amazon datasets. Feature extraction was done using n-grams. Support Vector Machine, k nearest neighbors and Random Forest classifiers were used to find sarcasm.

Bouazizi M. et al. [20] proposed a pattern based approach for detection of sarcasm. Sets of features were outlined which covered the different types of sarcasm. Tweets were used as dataset and the outcome of each and every tweet was then used for classification of polarity of the tweets.

Dave D. et al. [16] used reviews of Hindi language to detect sarcasm. Term frequency Inverse document frequency was used for word embeddings and Support Vector Machine was used for classification.

4.1.5 Deep Learning

Sharma K. et al. [46] used GLOVE word embeddings and Word2Vec word embeddings to convert text to vectors. The datasets were 2 publicly available datasets. One dataset had long sentences while the other dataset had short sentences. LSTM technique was used to detect sarcasm.

Kavitha K. et al. [47] did hyper parameter tuning to detect the presence of sarcasm in text. A combination of CNN and RNN was used for classification. GLOVE word embeddings were used for text to vectors and publicly available datasets were used.

Goel P. et al. [48] made an ensemble model to detect sarcasm. GRU, Bi-LSTM and CNN techniques were used. Datasets were picked from social media. The model sometimes erred on detecting false negatives.

4.1.6 Transformers

Wen Z. et al. [13] proposed a model where at word level, sememe knowledge was used and at sentence level, auxiliary knowledge for understanding the context of sarcasm. BERT was used to input this knowledge to the model.

Muaad A. et al. [31] worked on finding sarcasm in Arabic texts. 2 datasets were used for the research. AraBERT classifier was used to find sarcasm in the texts. The model was also used to find misogyny in the texts.

Savini E. et al. [33] used pre trained BERT models to detect sarcasm in texts. Intermediate task transfer learning was used to boost the performance.

Sharma D. et al. [49] used fusion of multiple models to detect sarcasm. Three publicly available datasets were used. BERT, LSTM-encoder and universal sentence encoder were used.

4.2 Techniques used in this study to detect Sarcasm

4.2.1 Naïve Bayes

The Naïve Bayes classifier is a machine learning algorithm based on Bayes theorem. It assumes that every feature in the classification is independent of the other. It is based on the principle of probability. An instance is classified belonging to a particular class using Naïve Bayes classifier.

Conditional probability of A given B indicates that event B has already occurred and we are finding the probability of A. It is denoted as P (A| B).

When we have multiple features b_1, b_2, b_3….b_n and we are finding if A belongs to a particular class, we denote it by P (A| b_1, b_2, b_3….b_n) and it is given by the formula:

$$P(A|\mathrm{b1}, \mathrm{b2}, \mathrm{b3}, \ldots . \mathrm{bn}\ ) = \frac{P(A) * P(\mathrm{b1}, \mathrm{b2}, \mathrm{b3}, \ldots . \mathrm{bn}\ |\mathrm{A})}{P(\mathrm{b1}, \mathrm{b2}, \mathrm{b3}, \ldots . \mathrm{bn})}$$

There are different types of Naïve Bayes models:

a. Gaussian: Used for continuous data.
b. Multinomial: Used for discrete data when there are multiple possibilities.
c. Bernoulli: Used for discrete data when there are only two possibilities.

It works comparatively faster than other machine learning algorithms. It works well on high dimensional data as well.

4.2.2 Support Vector Machine

Support Vector Machine is a classifier for classification & regression tasks. In Support Vector Machine, the main task is to build a hyperplane which can differentiate between two or more classes for all data points in the dataset. For a binary classification problem, the hyperplane can be a single line and the line acts as a partition between the two classes. For multi classification problem, the dimension of the hyperplane is dependent on the number of classes to be split into. The equation for the hyperplane is given by the formula:

$$v^T y + s = 0$$

In the equation, v represents the vector of the hyperplane, T denotes the transpose of vector v, y represents the input vector and s denotes the bias or the offset.

There are two types of Support Vector Machine models:

a. Linear SVM: When the data points in the dataset can be split into equal classes by a straight line/hyperplane, it is called as Linear SVM.
b. Nonlinear SVM: When the data points in the dataset cannot be split into separate classes by a straight line/hyperplane, it is called as Nonlinear SVM.

For high dimensional data, SVM is a good technique for classification but takes a lot of time to train the model.

4.2.3 Random Forest

Random Forest is an ensemble classifier that has a lot of decision trees. Each tree works on different parts of a dataset and predicts the output. The cumulative

average of each tree's opinion is taken so as to improve the performance metrics of that dataset. It is based on the assumption that instead of relying on only one output, the majority opinions of each tree in the Random Forest are considered to decide the final outcome.

The classifier works in two stages:

a. Selecting a random number of points from the training part of the dataset.
b. Depending on the number of data points, the decision trees are built.
c. The count of decision trees is decided.
d. This process is repeated and if there are any new points, the class is predicted depending on the count of majority opinions of the individual decision trees.

This classifier suffers from over fitting and hence, an optimal choice of number of trees has to be chosen depending on the size of the dataset.

4.2.4 Logistic Regression

The Logistic Regression classifier is used when there is a binary choice to be done. It predicts the outcome of a particular value/instance on probability and classifies it belonging to a particular class. The sigmoid function is used to find the probability value.
The sigmoid function helps in finding the probability by using the formula:

$$s = \frac{1}{1 + e^{-r}}$$

The input given to the sigmoid function is 'r' and the probability is stored in variable 's'.

The equation for Logistic Regression is:

$$P(z) = \frac{1}{1 + e^{-w.z+b}}$$

In the equation, z is the input value, P (z) is the outcome of the logistic regression classifier, b is the bias & w is the weight.

There are two types of Logistic Regression:

a. Binomial: It is used when there are only 2 classes of outcomes.
b. Multinomial: It is used when there are more than 2 classes of outcomes.

This classifier is easy to implement and is used when there are two or more categories of classes. It assumes linearity between the dependent and the independent variable which may not always be the case.

4.2.5 Gradient Boost

Gradient Boost classifier trains the model in a sequence and each upcoming model improvises on the previous one. The rationale behind Gradient boost is to combine multiple learners and convert those who are weak into strong ones by improvising. The improvisation can be done by reducing the value of the loss function using Gradient descent.

Initially a base model is made for predicting the output for a given set of input/inputs.

The initial step is written mathematically as follows:

$G_0(x) = \text{avg min } \Sigma\, L\,(y_i\,,\, \mu)$

L denotes the loss function, μ is the prediction and “avg min” is to find a value due to which the value of the loss is as less as possible.

The goal ahead is to reduce the value of loss as far as possible which can be done using differentiation to the loss function and equating it to zero. The difference between the prediction and the observation is done. Once completed, the outcome of every leaf in the model of our decision tree is found out and the average is taken as the final outcome.

4.2.6 Decision Tree

The decision tree classifier takes decisions on outcomes by finding the relations between variables in the dataset. Every decision tree has a root node, internal nodes, branches and leafs. The root node which is the starting node makes an initial decision. Each internal node has further branches and the internal nodes represent the decisions. The branches show the outcome of the decisions. The leafs represent the final prediction.

While creating a decision tree, the best attribute among the selected attributes is chosen. Entropy/Information Gain can be used for choosing the best attribute. Depending on the selected attribute, the whole dataset is split into multiple parts. This process is repeated over and over again till we reach the leaf node and classify the outcome in a particular class.

Decision trees suffer from over fitting. This can be resolved by pruning. Pruning is a technique to remove internal nodes which have little or less effect on the outcome of a particular instance.

4.2.7 k nearest neighbor (kNN)

The k nearest neighbor classifier is used to classify data points into groups for a particular feature. Once the groups are made, a new input is given to the model and it classifies it in a particular group/class.

The classifier works on finding the k nearest neighbors using a distance metric like the Euclidean distance. After the Euclidean distance is calculated, the group/class of the new data point is determined depending on the proximity of the groups around it.

The Euclidean distance is given by the formula:

$$distance = \sqrt{\Sigma\,(xj - Xij)^2}$$

The formula finds the Euclidean distance between two data points in the same hyper plane.

The right choice of value is necessary for the variable k. If the dataset is noisy and outliers are there, a higher value of k is preferred. Cross validation can also be used to find the right value of k.

The classifier suffers from over fitting and high dimensionality.

4.2.8 Stochastic Gradient Descent

The Stochastic Gradient Descent classifier works iteratively by changing the values of the parameters for reducing the loss incurred while doing the training. At any given iteration, a random set of values are picked from the dataset which helps in creating randomness about the values being trained at any given iteration. It is used in conjunction with other machine learning techniques for finding the optimal solution. A cost function is calculated at the end of each iteration. The cost function value is the difference between the predicted value and the actual

value given. By doing many iterative rounds, the loss function value is minimized and the nearer we go to the optimal value.

In stochastic gradient descent to obtain the value of the global minima, the cost function is calculated for any randomly chosen point from the dataset.

The cost function is given as

Cost function = $(a - g(b))^2$

where a is the value of the randomly chosen point and g(b) is the predicted value for the same data point.

The stochastic gradient classifier suffers from the problem of getting to the optimal value at a very slow rate.

4.2.9 Long Short Term Memory (LSTM)

LSTM technique was invented as an upgrade to the existing RNN technique. RNN suffers from the problem of learning dependencies from a long distance point of view. In LSTM technique, there is an additional cell given to store the dependencies from a long distance point of view.

This additional cell is controlled by three gates:

a. Input gate: The input gate decides what kind of information has to be stored in the additional cell.

b. Forget gate: The forget gate keeps vital information in the additional cell and discards what is not vital.

   The equation for the forget gate is given as

$$g_t = \delta\,(w_g\,(y_{t-1}, z_t) + c_g$$

Where

w_g represents the weight matrix of the forget gate

(y_{t-1}, z_t) represents the joining of the previous hidden state & the current input

c_g represents the bias

δ Is the sigmoid function.

c. Output gate: The output gate decides what kind of information is sent out from the additional cell.

LSTM's suffer from the problem of over fitting as well as require more time to train and learn.

4.2.10 Recurrent Neural Network (RNN)

RNN technique falls under deep learning in which the output of the predecessor is given as input to the next step. It has a hidden layer which records information in a sequence. To reduce the complexity, the same parameters are used for each input. It differs from other deep learning techniques with the virtue that the same weights are carried across the entire system.

The equation for the current state is given as

$$c_t = g(c_{t-1}, y_t)$$

Where

c_t represents the current state

c_{t-1} represents the previous state

y_t represents the input state

The output is calculated as

$$o_t = w_o a_t$$

where

o_t represents the output

w_o represents the output layer weight

a_t represents the activation function

It suffers from the vanishing gradient & the exploding gradient issue.

4.2.11 Convolution Neural Network (CNN)

CNN falls under deep learning and is used primarily for image classification. It is made as an extension to the original artificial neural networks. The CNN model contains the following layers:

a. Input layer

   It is used to give input to the system.

b. Convolution layer

   It is used to extract features from the text. Filters are applied at this layer after gathering data from the input layer. Feature maps are generated at this layer.

c. Pooling layer

   It is used to downsize the dimensions of the text. It is also used to reduce the problem of over fitting.

d. Fully connected layer

   It is used to finalize the outcome of the review by taking the input from the previous layer.

CNN suffers from over fitting & also requires a lot of annotated data for good performance.

4.2.12 Global vectors for Word Representation (GloVe)

GloVe is an unsupervised algorithm to find word embeddings of text. The model is made using a co-occurrence matrix of pairs of words. It was created so that the semantics between the different words in the dataset could be preserved. The values in GloVe are in a continuous vector space such that the semantics between the words are preserved. In this model, lot of dense embeddings is available. Every word in the dataset can be expressed in around 100 dimensions in word embeddings.

GloVe creates a word matrix for every word and relationship of that word with every other word in the dataset is expressed with a numerical decimal value. Optimization step is done after the matrix creation to create a dense vector for every word.

4.2.13 Bi-directional LSTM (BiLSTM)

Bi-directional LSTM technique is used in neural networks for a sequence model. It consists of 2 LSTM layers. One layer is for the input to be taken in the forward direction & the other for the processing backwards. The advantage of doing it is that the relationships between all the words can be understood in the forward as well as backward direction.

Two unidirectional LSTM's are used in Bi-directional LSTM technique. It can be considered as two different LSTM networks, with one going in the forward direction & the other in the backward direction. The LSTM going in the forward direction gets all the input in a sequential manner while the LSTM in the backward direction gets it in a reverse way. Both the LSTM's give a probability vector as an output & the final output is the combination of both the probability vectors.

It is represented by the equation

$$v_n = v_n^f + v_n^b$$

where

v_n = final outcome

v_n^f = forward outcome

v_n^b = backward outcome

It takes a lot of time to train and is quite slower compared to other neural networks.

4.2.14 Bidirectional Encoder Representations from Transformers (BERT)

Majority of the neural networks focus on understanding patterns from the data in a sequential manner, either left to right or right to left or both directions. The BERT model is trained bi-directionally and helps in recognizing context and the flow of the text in the dataset.

BERT uses transformers while training on the dataset. The BERT architecture uses an encoder for taking input and followed by a decoder which will predict the output. The advantage of using a transformer is that it can pick up the context of the entire review/sentence in one go rather than sequentially. It learns the context of a particular word in terms of the entire sentence/ review rather than a few words to its left or right.

The BERT model has two stages:

a. Training on large dataset (labeled/unlabeled) to understand the context and generate word embeddings.
b. Tuning the model on the labeled dataset for accurate predictions.

BERT adds a special layer on top of the encoder to learn prediction for the given sentence/review. The output from this layer is combined with the contextual word embeddings to get large dimensions for each word. SoftMax activation function is used to find the probability of each word expected in the outcome. The loss function is then used and the value between the predicted outcome and the expected outcome is calculated. It focuses mainly on the outcome words rather than all the words in the dataset at this stage.

The number of layers in the encoder stage and the decoder stage depends on the task to be performed. The "transformers" module of Python is used for BERT model.

It takes a lot of time to train the BERT model which is a disadvantage of the model.

4.2.15 Ensemble model (Naïve Bayes, Stochastic Gradient Descent & Logistic Regression)

An ensemble model of Naïve Bayes, Stochastic Gradient Descent and Logistic Regression was made. Hyperparameters were used to train the model with different combinations. N-grams were used to find the context between different words in the review. Smoothing parameters were used to manage zero counts. 7 fold cross validation was used. Logistic regression was the final estimator for deciding the outcome. The model used a stacking approach to learn optimal weights while training.

4.2.16 CB model (Convolution Neural Network & Bidirectional Encoder Representations from Transformers)

The hybrid model is a combination of CNN and BERT techniques. Initially, tokenization was done for all the sentences in the dataset using BERT transformer model. All the arrays were converted to tensors for processing. The model was built using 12 dense layers, 78 hidden layers, 12 heads and 110M parameters. Dropout layer was used to prevent over fitting. Sigmoid function was used as an activation function. Adam optimizer was used for convergence and to improve the training speed. For managing the loss, binary cross entropy function was used. For CNN, pad sequences were used to confine the length of each sentence to a pre-defined limit. Single dimensional convolutional layer is used for processing sequences. Fully connected dense layer is used.

4.3 Methodology to detect Sarcasm

4.3.1 Data collection

Data has been collected from the Kaggle website. “Headline news” is the first dataset. It has 28619 reviews. In this dataset, there are 3 fields. The first field is the label field which indicates if the particular review has sarcasm or not. The second column contains the review itself. Based on this review, is the value of the first column. The third column contains the link of the article from where the particular review was selected from. “Reddit reviews” is the second dataset. It has 80000 reviews. In this dataset, there are 10 fields. The first field is the label field which indicates if the particular review has sarcasm or not. The second column contains the review itself. Based on this review, is the value of the first column. The third column states the name of the author for the review. The fourth column states the domain of the review. The fifth, sixth and seventh column in the review state about the scores and the likes and dislikes for the particular review. The eighth column states the date and time the review was posted. The last column states if there was a parent review on which the Reddit user has posted the review.

The reason behind choosing these datasets were that they were publicly available as well as other researchers had done sarcasm detection tasks on these datasets. By choosing these datasets, a direct comparison could be done to judge the performance of our proposed model.

4.3.2 Pre-processing

The different steps in pre-processing are as follows:

4.3.2.1 Tokenization

Using the process of tokenization, a sentence/review can be split into individual tokens (words). This is essential as it creates a split up of the words into individual words and makes it easy to understand the structure of the sentence/review. Tokenization can be done by splitting the whole review into individual words as well as individual sentences. The nltk library of python can be used for tokenization. The nltk library contains sent_tokenize method to split the review into sentences and word_tokenize method to split each sentence into individual words.

For example the review is “Take incredible photos and videos in stunning color and details with the Pixel cameras. The Pixel 7 Pro offers the best Pixel camera yet with an upgraded ultra wide lens allowing users to capture stunning close-ups in Macro. Both smartphones are powered by fast, reliable, secure AT&T 5G and built with your security in mind. The Pixel 7 Pro offers an immersive 6.7-inch display that intelligently adjusts for a smoother, more responsive performance”. The sent_tokenize method will split the review into individual sentences by using full stop (.) as a reference point. The sentences individually would be “Take incredible photos and videos in stunning color and details with the Pixel cameras”, “The Pixel 7 Pro offers the best Pixel camera yet with an upgraded ultra

wide lens allowing users to capture stunning close-ups in Macro", "Both smartphones are powered by fast, reliable, secure AT&T 5G and built with your security in mind", "The Pixel 7 Pro offers an immersive 6.7-inch display that intelligently adjusts for a smoother, more responsive performance". Further, by using word_tokenize method each sentence will be split into individual words by using whitespace as a reference point.

4.3.2.2 Removal of stop words

Stop words are unnecessary words from the model point of view which depends on the task to be done. Commonly found words such as "in", "an", "am", etc. do not contribute to the task of sarcasm. The stopwords module is available in the corpus library of nltk.

4.3.2.3 Removal of null values and unwanted symbols

Cleaning of text is necessary to retain only those words which are essential in determining the presence of sarcasm in text. All digits, brackets, symbols like "*", escape sequences, etc. are removed from the dataset and preferably the text is converted to lower case.

4.3.2.4 Lemmatization

Lemmatization, which as an alternative to stemming, is used to convert a word to its basic stem. Stemming sometimes gives a word which does not have a meaning. By doing lemmatization, the word we get is a complete English word. For example, the word "discovering" after doing stemming will be "discoveri" while using lemmatization will convert it to "discovery". In the stem library of nltk, WordNetLemmatizer is available to do the process of lemmatization.

4.3.3 Word embeddings

Word embeddings are needed for machines to understand the reviews/sentences passed to it. The transformers module in Python is needed to convert text to word embeddings in BERT. A random seed value is chosen to get good quality embeddings and it generates the embeddings randomly. The “bert-base-uncased” is used to convert all uppercase text to lowercase text. Tokenization takes place to split the sentence into singular words with each token in turn converted to an ID. Special tokens SEP and CLS are also added to the tokenized text. To separate sentences in single input sequence, SEP token is used. For every input sequence, a CLS token is added at the beginning. An attention score of 1 is preferred while doing this process which indicates that the context of the sentence has been captured in the word embeddings. These tokens are then passed to the BERT model to generate word embeddings.

4.3.4 Splitting dataset

The dataset is cut into two sets of training and testing. The training set is used to train the model on the given dataset. 80% of the dataset is used to train the model. The remaining 20% is used to test the model trained and check the performance of the model. The sklearn library of Python is used to split the dataset into training and testing. The function “train_test_split” is used to split the dataset into training and testing sets. The parameters include the name of the pre-processed dataset/data frame, the size of the testing set, the size of the training set, a random state, shuffling parameter and the way to split the dataset.

4.3.5 Methodology

The necessary libraries for doing the task of sarcasm detection are imported like numpy, pandas, transformers, keras, etc. We tokenized all the individual reviews in the dataset using pre-trained BERT transformer model. Pre-processing techniques such as removal of stop words, symbols, etc. were done. The arrays were converted to tensors. The model was built using pre-trained BERT transformer model with 12 layers, 768 hidden layers, 12 heads and parameters. Dense layer was used for linearity followed by Dropout layer for reducing over fitting problem. Relu activation function was used to understanding the complex patterns in the data. The dataset was split into a ratio of 80:20 for training and testing. A batch size of 20 was chosen and the model was trained on the dataset. The model was then tested on the testing data and the performance of the model was judged on the evaluation metrics of accuracy, precision, recall and f-score.

4.3.6 Testing of the model

The testing set comprising of 20% of the dataset is used to check the performance of the model. It is necessary to test the model trained so as to judge its performance against some new data rather than testing on the training data itself. The data in the testing dataset is different from the training dataset. Another important point to be considered is the problem of over fitting. When the model understands & learns the training data properly but does not react properly to the testing data or any new data, it is called as over fitting. To overcome the problem of over fitting, the dataset is split into training and testing sets.

4.3.7 Performance of the model

The evaluation metrics are used to measure the quality of an algorithm/method/model in Natural Language Processing. Different models can be compared and judged on the basis of the evaluation metrics. The most prominent evaluation metrics are:

4.3.7.1 Accuracy

It is calculated as the ratio of the correctly predicted samples to the total number of samples. The accuracy of the model determines how well the model performs. It tells us whether the model is trained properly and how it will perform.

Accuracy = (True Positive + True Negative) / (True Positive + True Negative + False Positive + False Negative)

4.3.7.2 Precision

It is calculated as the ratio of number of samples belonging to positive class to the true positive samples and false positive samples.

Precision = True Positive / (True Positive + False Positive)

4.3.7.3 Recall

It is calculated as the ratio of correctly predicted positive samples to the number of correct positive samples and wrong negative samples. The recall of the model determines the number of correctly predicted positive samples to all the positive labels.

Recall = True Positive / (True Positive + False Negative)

4.3.7.4 F-score

It is calculated as the weighted average of precision and recall. It is calculated to take both false positives and false negatives into account. It is best to be used as it

is a combination of precision and recall. It is particularly useful on an unbalanced dataset.

F-score = (2 * precision * recall) / (precision + recall)

4.4 Results for Sarcasm Detection

Machine learning techniques and deep learning techniques have been used in the research to detect Sarcasm. An ensemble model has also been used for the same. The hybrid framework of CB technique gives the best performance scores across both the datasets used in this study for Sarcasm detection. Detailed analysis of results is explained in Chapter 6 for Sarcasm detection.

Chapter 5

Implementation of EDS Framework for Detection of Implicit Aspects

5.1 Techniques used to detect Implicit Aspects

Initial research on detecting implicit aspects was using association rule mining, co-occurrence matrix, ontology, clustering, dependency parsing and so on. However, more recently researchers have adopted deep learning techniques such as encoder-decoder, etc. to detect implicit aspects.

We discuss a few of the techniques here:

5.1.1 Co-occurrence Matrix:

Co-occurrence matrix is a very popular tool used to find the relation between different words in a dataset. It can depict the similarities and differences between words in a dataset. In its prime form, the co-occurrence matrix will use frequency of the terms which are matching within a specified window size. In co-occurrence matrix, we include all the words in the dataset which leads many times to a sparse matrix. Using pre-processing techniques like removing stop words, we can reduce the density of the matrix which leads to better accurate results as well as faster processing and execution.

In our study, we have extracted nouns and adjectives for the co-occurrence matrix. For each review, we have extracted the nouns and adjectives and appended them to the co-occurrence matrix.

5.1.2 Rule Based method:

Association rule mining is finding patterns and relationships between words in a dataset. In data mining as well as Natural language processing, rules are used to carry out tasks such as recommendation systems, market basket analysis, etc. We have the option to make our own customized rules for implicit aspect identification.

In our study, we have extracted the aspects and their opinions by taking the nouns and adjectives from the reviews. We have included these nouns and adjectives in lists of python and then depending on the outcomes (implicit aspects) of the model, we have filtered out everything except for the aspects to detect the correct implicit aspect(s).

5.1.3 Encoder-Decoder:

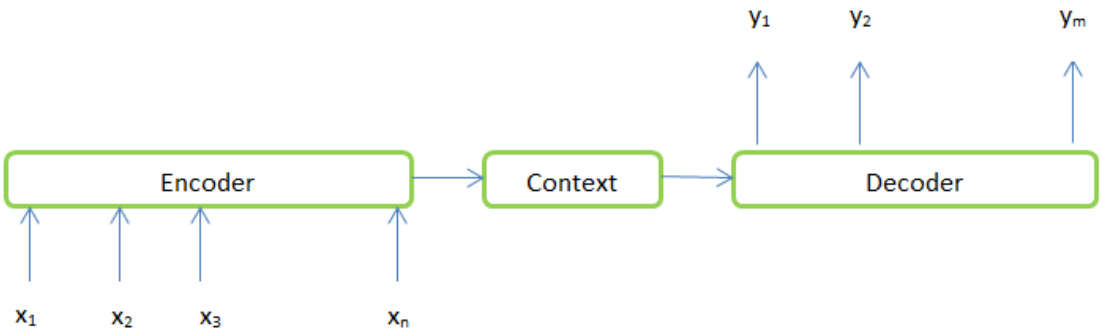

Figure 5.1 Encoder-Decoder Architecture

The encoder-decoder model is a well-known technique to detect words in a sequence. An important part of the encoder decoder model is that the lengths of the inputs and outputs can be different. At the encoder stage, the input is fed word by word to the encoder. The encoder contains a stack of several recurrent units, where each unit takes a single word at a time of the input given, stores

information about it and passes it forward. With reference to Figure 1., the input sequence is passed as x_1, x_2, x_3,…x_n. Each of this input sequence goes through multiple recurrent units in the encoder where each unit has a hidden state to store relevant information about the input words.

The hidden state h_j for each word is calculated as follows:

$$h\_j = f(w^h * h\_j_{-1} + w^{(hx)} * x_j)$$

In the above formula, the appropriate weights are applied to the previous state h_j_{-1} and the input x_j to generate the hidden state h_j.

The encoder decoder vector is the final hidden state from the encoder of the model. This vector keeps all context information from all the input given to the encoder. It in turn will be the first hidden state for the decoder of the model.

The decoder contains a stack of several recurrent units where each produces an output y_j for each step. Each unit of the decoder gets a hidden state from the previous unit and produces an output as well as a hidden state. The output is computed by using softmax activation function by multiplying the weights with the value of the hidden state.

5.1.4 Supervised Learning:

For this study, we have used a neural network to detect the implicit aspects. We have used a sequential dense layer model with relu activation function, sigmoid function and softmax activation function to classify and predict the implicit aspects. Categorical cross entropy function for loss and Adam optimizer. We split the dataset into 70:30 ratio of training and validation. We made a batch size of 50 with 1000 epochs.

5.2 Methodology to detect Implicit Aspects

5.2.1 Data collection

Data was extracted from the “X” website by querying in the term “mobile reviews”. 13918 records were extracted using snscrape library of Python. In this dataset, there are 4 fields. The first column contains the User ID of the person posting the tweet. The second column contains the date the tweet was posted. The third column contains the number of likes the tweet has received. The fourth column contains the tweet itself. Many of the reviews were available as links. So, the links were accessed and reviews from the link source were copied into the dataset. Additionally, we have made a customized dataset on mobile reviews by posting a questionnaire and getting reviews from different people online as well as offline. Around 1000 reviews were collected from the questionnaire responses. The questionnaire has columns such as “Brand of Mobile”, “Model Name”, columns on multiple features like camera, video quality, memory, etc. The last column contains a summarized review about the mobile phone for each user. In addition to this, we have manually annotated 200 mobile reviews with their implicit aspects. It was necessary to do manual annotation as the original reviews are unstructured and unlabeled. By doing manual annotation by experts in English literature, we have included neural network in our framework as a back up to the encoder decoder technique. For the task of Implicit Aspect Detection, unlabeled data was necessary as there could be one or multiple implicit aspects per sentence/review. The reason behind choosing Twitter data as well as a customized questionnaire was that our dataset could be unique as well as our model could work on existing data as well as any live data given for prediction.

5.2.2 Pre-processing

The different steps in pre-processing are as follows:

5.2.2.1 Tokenization

Using the process of tokenization, a sentence/review can be split into individual tokens (words). This is essential as it creates a split up of the words into individual words and makes it easy to understand the structure of the sentence/review. Tokenization can be done by splitting the whole review into individual words as well as individual sentences. The nltk library of python can be used for tokenization. The nltk library contains sent_tokenize method to split the review into sentences and word_tokenize method to split each sentence into individual words.

For example the review is “Take incredible photos and videos in stunning color and details with the Pixel cameras. The Pixel 7 Pro offers the best Pixel camera yet with an upgraded ultra wide lens allowing users to capture stunning close-ups in Macro. Both smartphones are powered by fast, reliable, secure AT&T 5G and built with your security in mind. The Pixel 7 Pro offers an immersive 6.7-inch display that intelligently adjusts for a smoother, more responsive performance”. The sent_tokenize method will split the review into individual sentences by using full stop (.) as a reference point. The sentences individually would be “Take incredible photos and videos in stunning color and details with the Pixel cameras”, “The Pixel 7 Pro offers the best Pixel camera yet with an upgraded ultra wide lens allowing users to capture stunning close-ups in Macro”, “Both smartphones are powered by fast, reliable, secure AT&T 5G and built with your security in mind”, “The Pixel 7 Pro offers an immersive 6.7-inch display that intelligently adjusts for a smoother, more responsive performance”. Further, by using word_tokenize method each sentence will be split into individual words by using whitespace as a reference point.

5.2.2.2 Removal of stop words

Stop words are unnecessary words from the model point of view which depends on the task to be done. Commonly found words such as “in”, “an”, “am”, etc. do not contribute to the task of sarcasm. The stopwords module is available in the corpus library of nltk.

5.2.2.3 Removal of null values and unwanted symbols

Cleaning of text is necessary to retain only those words which are essential in determining the presence of sarcasm in text. All digits, brackets, symbols like “*”, escape sequences, etc. are removed from the dataset and preferably the text is converted to lower case.

5.2.2.4 Lemmatization

Lemmatization, which as an alternative to stemming, is used to convert a word to its basic stem. Stemming sometimes gives a word which does not have a meaning. By doing lemmatization, the word we get is a complete English word. For example, the word “discovering” after doing stemming will be “discoveri” while using lemmatization will convert it to “discovery”. In the stem library of nltk, WordNetLemmatizer is available to do the process of lemmatization.

5.2.2.5 Parts of Speech (POS)

In any given sentence, the words can be categorized into many parts. Some are adjectives, nouns, verbs, adverbs, pronouns, etc. They are called as POS. The python library of nltk has a function pos_tag to split the sentence/review into POS. After splitting, each and every word is categorized in some or the other POS.

It is of two types:

i. Rule based: A dictionary is used.
ii. Stochastic: Probability or count is used.

The categories after doing POS tagging are:

i. Adjective (JJ)
ii. Noun (NN)
iii. Verb (VBZ)
iv. Preposition (IN)
v. Adverb (ADV)
vi. Pronoun (PRON)

5.2.3 Word embeddings

Word2Vec technique is used to convert text to numeric content. Every word after conversion to numbers has a lot of dimensions in this technique. In the background, Word2Vec uses neural networks to convert text to numbers. It has an input layer, hidden layer and an output layer. It was invented by Google to convert text to numbers such that every word is expressed in high dimensions as well as the semantic information between different words is preserved. It has two architectures:

5.2.3.1 Continuous Bag of Words

This is used for prediction of a given word with other words given with it for a current set of words. The input layer has the words around the required word and the output layer gives the outcome of the word to be predicted. The hidden layer defines the number of dimensions with which the final predicted word will have.

5.2.3.2 Skip Gram

This is used for prediction of the words around a certain word for a given window size. The input layer has the current word and the output layer contains all those words which are related to the current word. The hidden layer defines the number of dimensions in which the current word in the input layer has to be expressed.

Word2Vec technique goes with the basic intuition that words which are similar to one another will have vectors similar to one another. The nltk module and gensim module of Python is used for generating Word2Vec word embeddings. The advantage of using Word2Vec word embeddings is that the connections between different words in the dataset are maintained and context is taken into consideration. Additionally, for datasets which are large in size and require heavy computation, Word2Vec runs efficiently.

5.2.4 Splitting dataset

The dataset is cut into two sets of training and testing. The training set is used to train the model on the given dataset. 80% of the dataset is used to train the model. The remaining 20% is used to test the model trained and check the performance of the model. The sklearn library of Python is used to split the dataset into training and testing. The function “train_test_split” is used to split the dataset into training and testing sets. The parameters include the name of the pre-processed dataset/data frame, the size of the testing set, the size of the training set, a random state, shuffling parameter and the way to split the dataset.

5.2.5 Methodology

Preprocessing techniques such as tokenization, removal of stop words, removal of unwanted symbols, removal of null values, conversion of words to lowercase and normalization was done initially on the dataset. Parts of Speech tagging was done

to extract the relevant nouns and adjectives from the reviews. We used word2vec model to convert text to vectors.

Keras library was used for the implementation of encoder-decoder technique. Long Short Term Memory technique is used to learn long term dependencies in the dataset. Sigmoid function is used as an activation function. Softmax activation function is used to convert the vectors in a probability distribution and we have normalized the data. Cross Entropy function is used for calculating the loss and Adam optimizer function to reduce the loss of the model. We used techniques such as co-occurrence matrix, encoder-decoder and rules on the unstructured data to understand the patterns and relationships in the data. The model was trained on the existing dataset and then any random input was fed to the model. The performance was noted down in the form of evaluation metrics. We have used supervised learning as an alternative to the encoder decoder model i.e. just in case, the encoder decoder model does not give the output, then supervised learning will predict the output.

5.2.6 Testing of the model

The testing set comprising of 20% of the dataset is used to check the performance of the model. It is necessary to test the model trained so as to judge its performance against some new data rather than testing on the training data itself. The data in the testing dataset is different from the training dataset. Another important point to be considered is the problem of over fitting. When the model understands & learns the training data properly but does not react properly to the testing data or any new data, it is called as over fitting. To overcome the problem of over fitting, the dataset is split into training and testing sets.

5.2.7 Performance of the model

The evaluation metrics are used to measure the quality of an algorithm/method/model in Natural Language Processing. Different models can be compared and judged on the basis of the evaluation metrics. The most prominent evaluation metrics are:

5.2.7.1 Accuracy

It is calculated as the ratio of the correctly predicted samples to the total number of samples. The accuracy of the model determines how well the model performs. It tells us whether the model is trained properly and how it will perform.

Accuracy = (True Positive + True Negative) / (True Positive + True Negative + False Positive + False Negative)

5.2.7.2 Precision

It is calculated as the ratio of number of samples belonging to positive class to the true positive samples and false positive samples.

Precision = True Positive / (True Positive + False Positive)

5.2.7.3 Recall

It is calculated as the ratio of correctly predicted positive samples to the number of correct positive samples and wrong negative samples. The recall of the model determines the number of correctly predicted positive samples to all the positive labels.

Recall = True Positive / (True Positive + False Negative)

5.2.7.4 F-score

It is calculated as the weighted average of precision and recall. It is calculated to take both false positives and false negatives into account. It is best to be used as it is a combination of precision and recall. It is particularly useful on an unbalanced dataset.

F-score = (2 * precision * recall) / (precision + recall)

5.3 Results for Implicit Aspect Detection

Unsupervised deep learning techniques have been used in the research to detect implicit aspects. The novel framework of EDS technique gives the best performance scores for Implicit Aspect detection. Detailed analysis of results is explained in Chapter 6 for Implicit Aspect detection.

5.4 Results from Input to Output for Implicit Aspect Detection

- **Input:** The phone is light but huge.

  Output: size, weight

  Here, the important words to consider are light and huge. The aspects are missing. The model gives the output as size for the opinion “huge” and weight for the opinion “light”.

- **Input:** i am bored with the silver look.

  Output: screen

  Here, the important word to consider is the silver look. The aspect is missing. The model gives the output as screen for the opinion “look”.

- **Input:** phone drains quickly.

  Output: battery

  Here, the important words to consider are the phone’s battery draining quickly. The aspect is missing. The model gives the output as battery for the opinion “drains”.

Chapter 6

Results and Discussion

6.1 Results and Discussion for Sarcasm Detection

TABLE 6.1 ANALYSIS OF MACHINE LEARNING TECHNIQUES ON HEADLINE NEWS DATASET FOR SARCASM

Methodology	**Accuracy**	**Precision**	**Recall**	**F-score**
Naïve Bayes	82	82	82	82
Stochastic Gradient Descent	81	81	81	81
k Nearest Neighbor	74	76	74	73
Logistic Regression	81	81	81	81
Decision Tree	59	69	59	50
Random Forest	78	78	78	77
Support Vector Machine	81	81	81	81
Gradient Boost	76	76	76	75
Ensemble model	81	80	81	81

All values in the table are in percentages

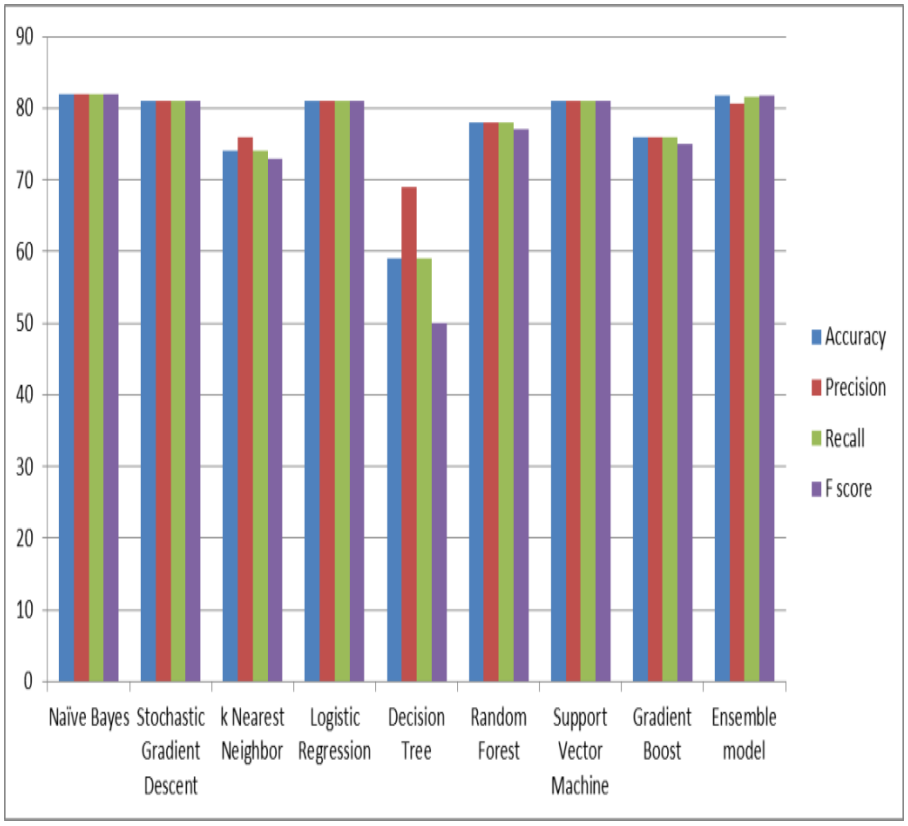

Figure 6.1 Analysis of Machine learning techniques on Headline news dataset for Sarcasm

TABLE 6.2 Analysis of Machine learning techniques on Reddit dataset for Sarcasm

Methodology	**Accuracy**	**Precision**	**Recall**	**F-score**
Naïve Bayes	64	64	64	64
Stochastic Gradient Descent	66	66	66	66
k Nearest Neighbor	61	63	61	52
Logistic Regression	65	65	65	65
Decision Tree	58	56	58	45
Random Forest	62	62	62	60
Support Vector Machine	63	63	63	58
Gradient Boost	61	63	61	55
Ensemble model	65	67	76	65

All values in the table are in percentages

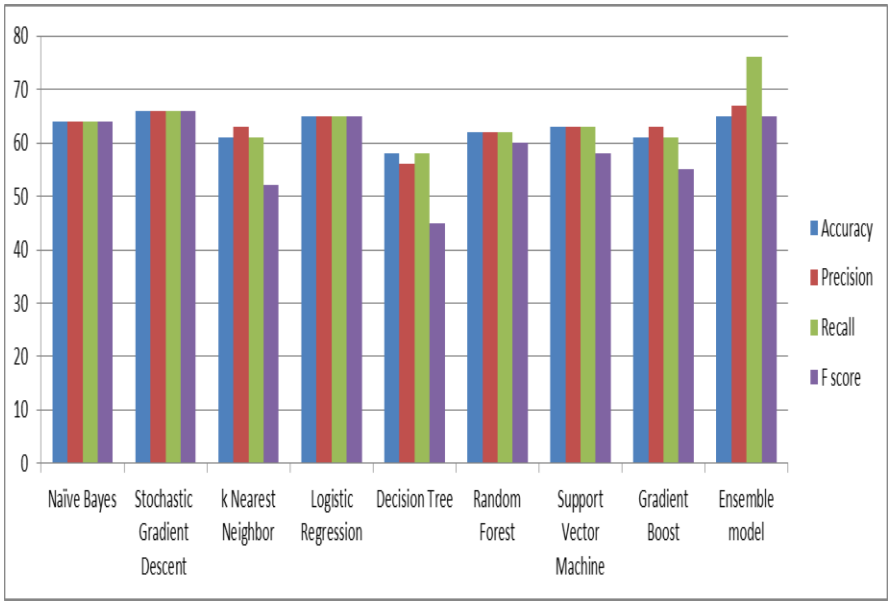

Figure 6.2 Analysis of Machine learning techniques on Reddit dataset for Sarcasm

TABLE 6.3 ANALYSIS OF DEEP LEARNING TECHNIQUES ON HEADLINE NEWS DATASET FOR SARCASM

Methodology	**Accuracy**	**Precision**	**Recall**	**F-score**
LSTM	73.07	71.68	64.10	67.68
LSTM + RNN	81	81	81	81
RNN	46.12	52	50	50
CNN	82	82	82	82
GLOVE	75.98	76	76	76
BiDirectional LSTM	70.63	66.92	65.73	66.32
BERT	92.73	93	93	93
CB Framework	**95.10**	**95**	**95**	**95**

All values in the table are in percentages

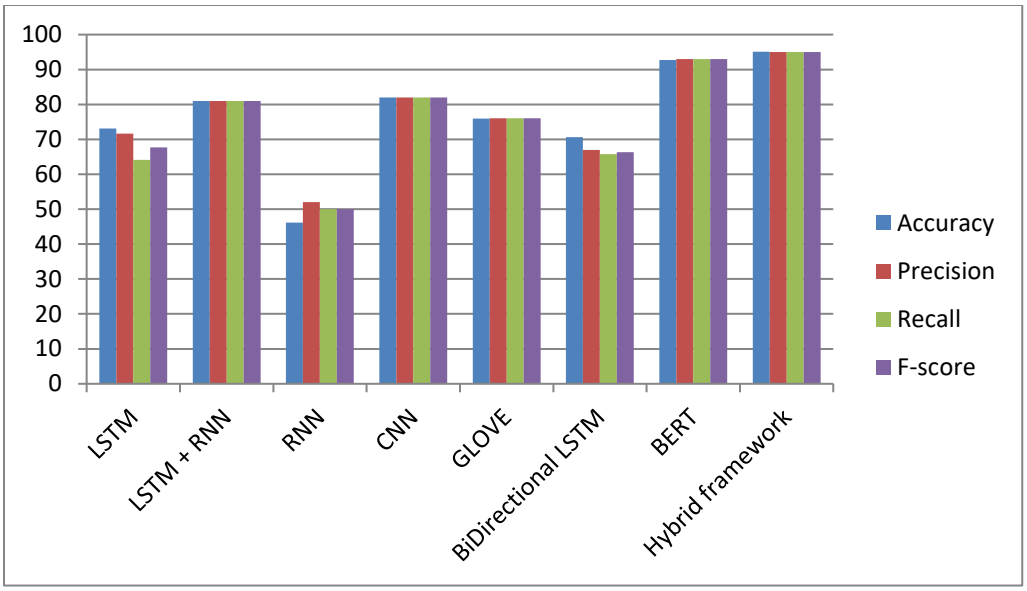

Figure 6.3 Analysis of Deep learning techniques on Headline news dataset for Sarcasm

TABLE 6.4 ANALYSIS OF DEEP LEARNING TECHNIQUES ON REDDIT DATASET FOR SARCASM

Methodology	**Accuracy**	**Precision**	**Recall**	**F-score**
LSTM	63	63	63	63
LSTM + RNN	58	58	58	58
RNN	30	47	30	50
CNN	62	62	62	62
GLOVE	64.70	66.03	62.64	64.29
BiDirectional LSTM	58.59	59.10	56.47	57.75
BERT	75	76	75	74
CB Framework	**93.81**	**94**	**94**	**94**

All values in the table are in percentages

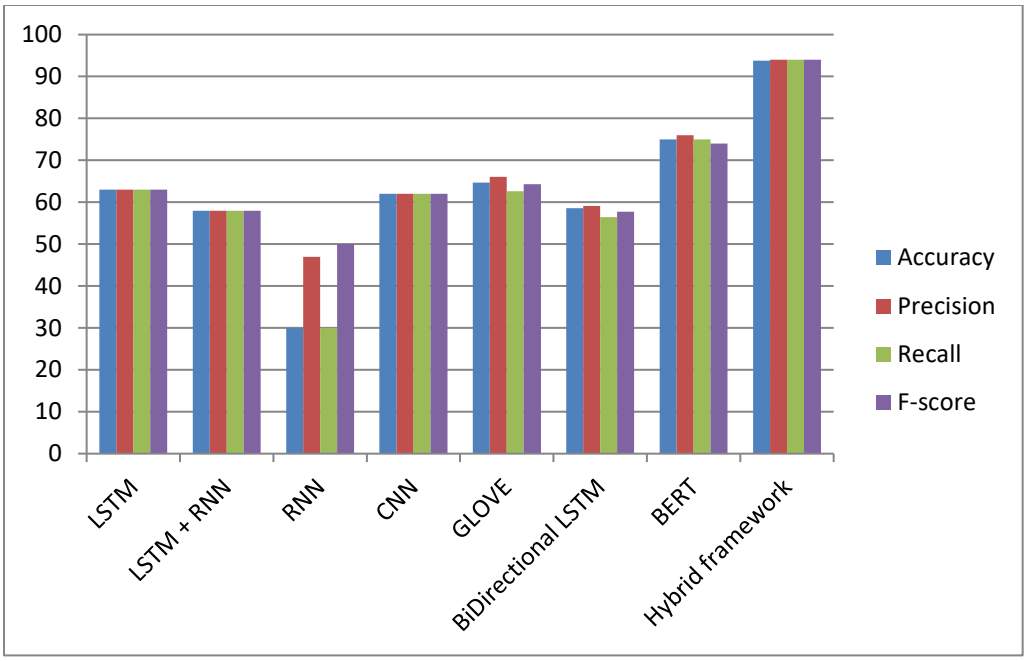

Figure 6.4 Analysis of Deep learning techniques on Reddit dataset for Sarcasm

TABLE 6.5 ERROR ANALYSIS OF TECHNIQUES USED IN THIS STUDY ON HEADLINES DATASET FOR SARCASM

Methodology	**Number of Epochs**	**Time taken**	**Loss function**
LSTM	30	2 hours	0.027
LSTM + RNN	30	5.5 hours	0.053
RNN	30	5.5 hours	0.071
CNN	100	1.5 hours	0.009
GLOVE	30	1 hour	0.037
BiDirectional LSTM	30	2 hours	0.405

BERT	100	5 hours	0.093
CB Framework	**100**	**10 hours**	**0.208**

TABLE 6.6 ERROR ANALYSIS OF TECHNIQUES USED IN THIS STUDY ON REDDIT DATASET FOR SARCASM

Methodology	**Number of Epochs**	**Time taken**	**Loss function**
LSTM	30	2 hours	0.056
LSTM + RNN	30	5.5 hours	0.180
RNN	30	5.5 hours	0.453
CNN	100	1.5 hours	0.022
GLOVE	30	1 hour	0.549
BiDirectional LSTM	30	2 hours	0.039
BERT	100	5 hours	0.560
CB Framework	**100**	**10 hours**	**0.363**

TABLE 6.7 PERFORMANCE COMPARISON OF CB FRAMEWORK WITH EXISTING MODELS ON HEADLINES DATASET FOR SARCASM

Name of Paper & Year	Method	Accuracy	Precision	Recall	F score
Harnessing Advanced Learning for Sarcasm Detection, 2024	Encoder	80.1	---	---	75.5
Sarcasm Detection in News Headlines with Deep Learning, 2024	BERT	88	---	---	---
Sarcasm Detection in News Headlines using ML and DL models, 2024	RNN	79	---	---	76
An Efficient Sarcasm Detection using Linguistic Features and Ensemble Machine Learning, 2024	Ensemble (Gradient Boost, Decision Tree, Random Forest)	93.75	---	---	---
Detect Sarcasm and Humor jointly by Neural Multi-Task Learning, 2024	RNN	---	79.34	79.88	79.61
Sarcasm Detection over Social Media Platforms Using Hybrid Ensemble Model with Fuzzy Logic, 2023	BERT + fuzzy logic	90.81	---	---	---

An Ensemble Model for detecting Sarcasm on Social Media, 2022	Glove + LSTM	88.9	91.1	88.67	89.87
Context-Driven Satire Detection With Deep Learning, 2022	SVM, KNN	---	91	90	91
Sarcasm Detection over Social Media Platforms Using Hybrid Auto-Encoder-Based Model	Encoders	90.8	---	---	---
CB Framework		**95.10**	**95**	**95**	**95**

TABLE 6.8 PERFORMANCE COMPARISON OF CB FRAMEWORK WITH EXISTING MODELS ON REDDIT DATASET FOR SARCASM

Name of Paper & Year	**Method**	**Accuracy**	**Precision**	**Recall**	**F score**
IdSarcasm: Benchmarking & Evaluating Language Models for Indonesian Sarcasm Detection, 2024	Large Language Model	--	--	--	62.7
Sarcasm Detection over Social Media Platforms Using Hybrid Ensemble Model with Fuzzy	BERT + fuzzy logic	85.38	--	--	--

Logic, 2024					
Intermediate-Task Transfer Learning with BERT for Sarcasm Detection, 2022	BERT	--	--	--	77.53
Affection Enhanced Relational Graph Attention Network for Sarcasm Detection, 2022	Graph Attention Network	75.8	75.8	75.8	75.8
Jointly Learning Sentimental Clues and Context Incongruity for Sarcasm Detection, 2022	Semantics	73.96	73.98	74.07	73.42
Detecting the target of sarcasm is hard: Really??, 2022	LSTM	71.5	--	--	--
Sarcasm Detection over Social Media Platforms Using Hybrid Auto-Encoder-Based Model, 2022	Encoders	83.92	--	--	--
An Effective Sarcasm Detection	Bidirectional LSTM with	71	--	--	71

Approach Based on Sentimental Context and Individual Expression Habits, 2022	SenticNet library				
Sarcasm detection using deep learning and ensemble learning, 2022	CNN + LSTM + GRU	81.13	--	--	--
CB Framework		**93.81**	**94**	**94**	**94**

The ML and DL algorithms are tested on two datasets. The first is the News Headlines dataset and the second is the text collected from the Reddit website. The results of testing the ML algorithms on the headlines dataset with the parameters of accuracy, precision, recall and f-score is mentioned in Table no. 6.1 and for the Reddit dataset in Table no. 6.2. The results of testing the DL algorithms on the headlines dataset with the parameters of accuracy, precision, recall and f-score is mentioned in Table no. 6.3 and for the Reddit dataset in Table no. 6.4.

From Table number 6.1 and Figure 6.1, for the Headline news dataset, we can conclude that Naïve Bayes classifier and the ensemble model gives the best performance. An accuracy of 82 and f-score of 82 is achieved among the machine learning algorithms.

From Table number 6.2 and Figure 6.2, for the Reddit dataset, we can conclude that Stochastic Gradient Descent classifier and the ensemble model gives the best performance. An accuracy of 66 and f-score of 66 is achieved among the machine learning algorithms.

In Table number 6.3 and Figure 6.3, we have done a comparative analysis on the news headlines dataset using DL techniques. The CB framework gives us a better performance score compared to the individual techniques of CNN & BERT with an accuracy score of 95.10% and f-score of 95% which is an improvement of 2% over the individual techniques.

In Table number 6.4 and Figure 6.4, we have done a comparative analysis on the Reddit dataset using DL techniques. The CB framework gives us a better performance score compared to the individual techniques of CNN & BERT with an accuracy score of 93.81% and f-score of 94% which is an improvement of 18% over the individual techniques.

In Table number 6.7, we have done a comparative analysis of our framework on the Headlines dataset with models and techniques of other researchers. The CB framework gives us a better performance score.

In Table number 6.8, we have done a comparative analysis of our framework on the Reddit dataset with models and techniques of other researchers. The CB framework gives us a better performance score.

Table 6.5 and Table 6.6 shows the error analysis for the techniques used on the Headlines dataset and the Reddit dataset.

From Table number 6.3, Table number 6.4, Table number 6.7 and Table number 6.8, it can be concluded that the CB framework gives the best performance.

6.2 Experimental Results and Analysis for Implicit Aspects

Table 6.9 Analysis of techniques for Implicit Aspect detection

Dataset	Methodology	Accuracy	Precision	Recall	F score
Twitter + Questionnaire	Co-occurrence matrix	64	13	11	13
	Rule method	66	79	79	77
	EDS Framework	**83**	**92**	**78**	**84**

All values in the table are in percentages

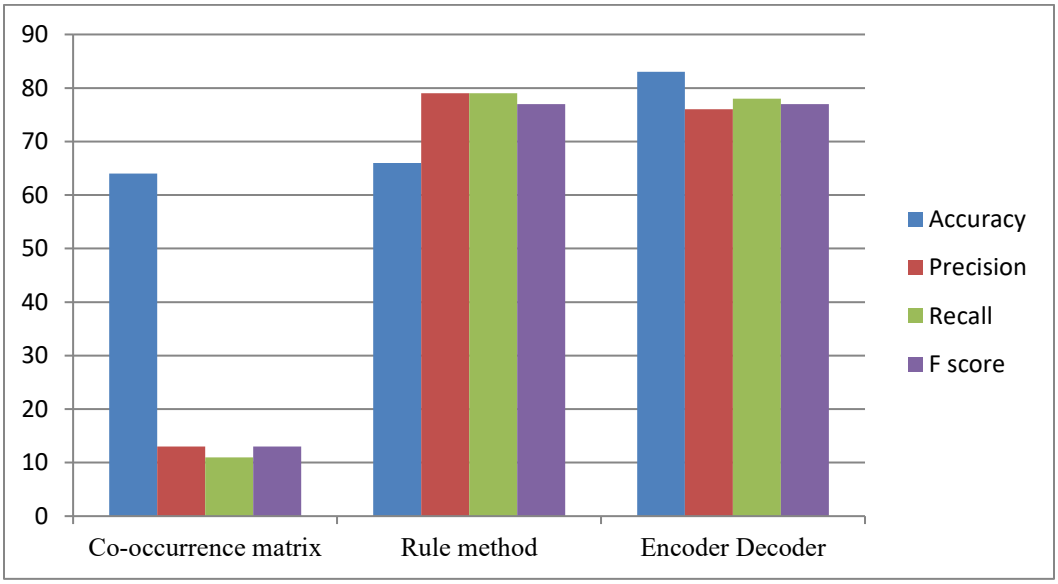

Figure 6.5 Analysis of techniques for Implicit Aspect detection

Table 6.10 Error Analysis of Techniques used in this study for Implicit Aspect Identification

Methodology	**Number of Epochs**	**Time taken**	**Loss function**
EDS Framework	1000	3 hours	0.059

Table 6.11 Performance comparison of EDS Framework with existing models for Implicit Aspect Detection

Name of Paper & Year	**Method**	**Accuracy**	**Precision**	**Recall**	**F score**
A novel context-based implicit feature extracting method	Opinion words and matrix	70	71	62	66
Implicit feature identification via co-occurrence association rule mining	Co-occurrence and Association	74	76	72	74
Extract product features in Chinese web for opinion mining	Clustering	76	79	70	74
An association-based unified framework for mining features and opinion words	Association	72	72	69	71
A classification-based approach for implicit feature identification	Machine Learning	76	82	68	74
Joint topic-opinion model for implicit feature extracting	Topic Modeling	78	82	68	74
An improved association rule mining approach to identification of implicit product aspects	Association	77	82	73	77
EDS Framework		**83**	**92**	**78**	**84**

In Table number 6.9 and Figure 6.5, we have compared the three techniques that we have used for detecting implicit aspects. The co-occurrence matrix technique gives an accuracy score of 64% and the rule based technique gives an accuracy of 66%. The encoder-decoder model gives the best accuracy score of 83% and f-score of 77%.

Table 6.10 shows the error analysis for the EDS technique for the Implicit Aspect Identification task.

In Table number 6.11, we have done a comparative analysis of our framework with models and techniques of other researchers for Implicit Aspect Identification. The EDS framework gives us a better performance score.

From Table number 6.9 and Table number 6.11, we can conclude that the EDS framework gives the best performance.

Chapter 7

Conclusion and Future Work

7.1 Conclusion and Future Work for Sarcasm Detection

Sarcasm detection in text is a challenge and it is difficult to spot the presence of sarcasm. In this study, we have used eight machine language classifiers and deep learning techniques to detect sarcasm. Based on publicly accessible datasets of Headline news and Reddit, we train and test the different ML and DL techniques. Preprocessing techniques such as removal of stop words, removal of null values, tokenization and normalization are done on both the datasets. Word embedding techniques have been used to convert text to vectors. We have split both the datasets in the ratio of 80:20 for training and testing, respectively. Performance metrics such as accuracy, precision, recall and f-score are used. For the Headline news dataset as well as Reddit dataset, Bidirectional Encoder Representations from Transformers technique gives the best performance with an accuracy of 92.73% and f-score of 93% on the Headline news dataset and an accuracy of 75% and f-score of 74% on the Reddit dataset. It is to be noted that the performance is better on the Headline news dataset as the headline contains the whole context of the topic while in the Reddit dataset there are a lot of reviews with less contextual knowledge.

Opinion mining is a vital task for majority of the businesses/organizations to get an assessment about their products and services. Social media is a place where people all over the globe discuss and review different products. People tend to post reviews sometimes sarcastically. It is necessary to find such sarcastic reviews and invert their polarity to get a clear view about the reviews. For this reason, we have used DL techniques of CNN, BERT & a hybrid framework of CNN + BERT to find sarcasm in two publicly available datasets of news headlines and Reddit. The CB framework gives better performance scores than the individual techniques.

The novelty of this research is that we have done a fine comparison analysis between the existing techniques for sarcasm detection and proposed a new CB framework to detect sarcasm.

For future work, an evolutionary approach could be used to detect sarcasm as newer ways are found by people to express sarcasm. Also, support for sarcasm detection in languages used across the world.

7.2 Conclusion and Future Work for detection of Implicit Aspects

Detection of implicit aspects in text is a challenge and it is difficult to spot. In this study, we have used unsupervised techniques to detect implicit aspects in mobile reviews dataset. Based on data scrapped from Twitter and our questionnaire, we train the unsupervised techniques to spot implicit aspects in our dataset. Preprocessing techniques such as removal of stop words, removal of null values, tokenization and normalization are done on the dataset. Word embedding techniques have been used to convert text to vectors. Performance metrics such as accuracy, precision, recall and f-score are used.

Encoder decoder technique gives the best performance with an accuracy of 83% and f-score of 77% on our dataset.

For future work, we would go ahead for a better evolutionary method than our existing techniques.

7.3 Limitations/Challenges for Sarcasm Detection and Implicit Aspects

The context is difficult to capture in too short sentences. The CB technique is better here compared to other techniques due to the pre-training of BERT to capture sarcasm clues. Culture-specific sarcasm is difficult to capture by the CB technique. Quotations/Emphasis are detected by the CB technique due to the CNN detecting the patterns in the review. Also, the CB technique captures contrastive sentiments in the review easily due to the attention mechanism and pattern recognition.

Lack of labeled datasets for the task of Implicit Aspect Detection is a challenge. Also, cross domain generalization task for detecting implicit aspects is a tedious task due to different domains having different aspects and their meanings.

References

[1] R. Bhalla and Amandeep, "A Comparative Analysis of Application of Proposed and the Existing Methodologies on a Mobile Phone Survey," *Commun. Comput. Inf. Sci.*, vol. 1206 CCIS, pp. 107–115, 2020, doi: 10.1007/978-981-15-4451-4_10.

[2] R. Bhalla and A. Bagga, "Opinion mining framework using proposed rb-bayes model for text classification," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 1, pp. 477–484, 2019, doi: 10.11591/ijece.v9i1.pp477-484.

[3] N. Wedjdane, R. Khaled, and K. Okba, "Better Decision Making with Sentiment Analysis of Amazon reviews," *Proc. - 2021 Int. Conf. Inf. Syst. Adv. Technol. ICISAT 2021*, 2021, doi: 10.1109/ICISAT54145.2021.9678483.

[4] M. P. Abraham, "Feature Based Sentiment Analysis of Mobile Product Reviews using Machine Learning Techniques," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 2, pp. 2289–2296, 2020, doi: 10.30534/ijatcse/2020/210922020.

[5] R. S. Jagdale, V. S. Shirsat, and S. N. Deshmukh, "Sentiment analysis on product reviews using machine learning techniques," *Adv. Intell. Syst. Comput.*, vol. 768, pp. 639–647, 2019, doi: 10.1007/978-981-13-0617-4_61.

[6] Y. Hegde and S. K. Padma, "Sentiment Analysis for Kannada using mobile product reviews: A case study," *Souvenir 2015 IEEE Int. Adv. Comput. Conf. IACC 2015*, pp. 822–827, 2015, doi: 10.1109/IADCC.2015.7154821.

[7] Z. Singla, S. Randhawa, and S. Jain, "Statistical and Sentiment Analysis," pp. 1–6, 2017.

[8] M. A. M. Salem and A. Y. A. Maghari, "Sentiment analysis of mobile phone products reviews using classification algorithms," *Proc. - 2020 Int. Conf. Promis. Electron. Technol. ICPET 2020*, pp. 84–88, 2020, doi: 10.1109/ICPET51420.2020.00024.

[9] F. Wu, Z. Shi, Z. Dong, C. Pang, and B. Zhang, "Sentiment analysis of online product reviews based on SenBERT-CNN," *Proc. - Int. Conf. Mach. Learn. Cybern.*, vol. 2020-Decem, pp. 229–234, 2020, doi: 10.1109/ICMLC51923.2020.9469551.

[10] P. Pankaj, P. Pandey, M. Muskan, and N. Soni, "Sentiment Analysis on Customer Feedback Data: Amazon Product Reviews," *Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019*, pp. 320–322, 2019, doi: 10.1109/COMITCon.2019.8862258.

[11] S. Yadav and N. Saleena, "Sentiment analysis of reviews using an augmented dictionary approach," *Proc. 2020 Int. Conf. Comput. Commun. Secur. ICCCS 2020*, pp. 0–4, 2020, doi: 10.1109/ICCCS49678.2020.9277094.

[12] Rekha and W. Singh, "Sentiment analysis of online mobile reviews," *Proc. Int. Conf. Inven. Commun. Comput. Technol. ICICCT 2017*, no. Icicct, pp. 20–25, 2017, doi: 10.1109/ICICCT.2017.7975199.

[13] Z. Wen *et al.*, "Sememe knowledge and auxiliary information enhanced approach for

sarcasm detection," *Inf. Process. Manag.*, vol. 59, no. 3, 2022, doi: 10.1016/j.ipm.2022.102883.

[14] Y. Du, T. Li, M. S. Pathan, H. K. Teklehaimanot, and Z. Yang, "An Effective Sarcasm Detection Approach Based on Sentimental Context and Individual Expression Habits," *Cognit. Comput.*, vol. 14, no. 1, pp. 78–90, 2022, doi: 10.1007/s12559-021-09832-x.

[15] S. S. Sonawane and S. R. Kolhe, "TCSD: Term Co-occurrence Based Sarcasm Detection from Twitter Trends," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 830–839, 2020, doi: 10.1016/j.procs.2020.03.422.

[16] A. D. Dave and N. P. Desai, "A comprehensive study of classification techniques for sarcasm detection on textual data," *Int. Conf. Electr. Electron. Optim. Tech. ICEEOT 2016*, pp. 1985–1991, 2016, doi: 10.1109/ICEEOT.2016.7755036.

[17] K. Sentamilselvan, P. Suresh, G. K. Kamalam, S. Mahendran, and D. Aneri, "Detection on sarcasm using machine learning classifiers and rule based approach," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1055, no. 1, p. 012105, 2021, doi: 10.1088/1757-899x/1055/1/012105.

[18] R. Afiyati, E. Winarko, and A. Cherid, "Recognizing the sarcastic statement on WhatsApp Group with Indonesian language text," *2017 Int. Conf. Broadband Commun. Wirel. Sensors Powering, BCWSP 2017*, vol. 2018-Janua, no. May, pp. 1–6, 2018, doi: 10.1109/BCWSP.2017.8272579.

[19] P. Katyayan and N. Joshi, "Sarcasm Detection Algorithms Based on Sentiment Strength," *Intell. Data Anal.*, no. June, pp. 289–306, 2020, doi: 10.1002/9781119544487.ch14.

[20] M. Bouazizi and T. Otsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter," *IEEE Access*, vol. 4, no. January, pp. 5477–5488, 2016, doi: 10.1109/ACCESS.2016.2594194.

[21] M. V. Rao and C. Sindhu, "Detection of Sarcasm on Amazon Product Reviews using Machine Learning Algorithms under Sentiment Analysis," *2021 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2021*, no. October, pp. 196–199, 2021, doi: 10.1109/WiSPNET51692.2021.9419432.

[22] A. Ashwitha, G. Shruthi, H. R. Shruthi, M. Upadhyaya, A. P. Ray, and T. C. Manjunath, "Sarcasm detection in natural language processing," *Mater. Today Proc.*, vol. 37, no. Part 2, pp. 3324–3331, 2020, doi: 10.1016/j.matpr.2020.09.124.

[23] R. Justo, T. Corcoran, S. M. Lukin, M. Walker, and M. I. Torres, "Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web," *Knowledge-Based Syst.*, vol. 69, no. 1, pp. 124–133, 2014, doi: 10.1016/j.knosys.2014.05.021.

[24] H. M. K. Kumar and B. S. Harish, "Sarcasm classification: A novel approach by using Content Based Feature Selection Method," *Procedia Comput. Sci.*, vol. 143, pp. 378–386, 2018, doi: 10.1016/j.procs.2018.10.409.

[25] S. Mukherjee and P. K. Bala, "Sarcasm detection in microblogs using Naïve Bayes and

fuzzy clustering," *Technol. Soc.*, vol. 48, pp. 19–27, 2017, doi: 10.1016/j.techsoc.2016.10.003.

[26] S. V. Oprea and W. Magdy, "Exploring author context for detecting intended vs perceived sarcasm," *ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, pp. 2854–2859, 2020, doi: 10.18653/v1/p19-1275.

[27] K. Sundararajan and A. Palanisamy, "Probabilistic Model Based Context Augmented Deep Learning Approach for Sarcasm Detection in Social Media," vol. 29, no. 6, pp. 8461–8479, 2020.

[28] P. Parameswaran, A. Trotman, V. Liesaputra, and D. Eyers, "Detecting the target of sarcasm is hard: Really??," *Inf. Process. Manag.*, vol. 58, no. 4, p. 102599, 2021, doi: 10.1016/j.ipm.2021.102599.

[29] D. V. P. Prabhavathy, "An intelligent machine learning ‐ based sarcasm detection and classification model on social networks," *J. Supercomput.*, no. 0123456789, 2022, doi: 10.1007/s11227-022-04312-x.

[30] A. M. A. Barhoom, B. S. Abunasser, and S. S. Abu-naser, "Sarcasm Detection in Headline News using Machine and Deep Learning Algorithms," vol. 6, no. 4, pp. 66–73, 2022.

[31] A. Y. Muaad *et al.*, "Artificial Intelligence-Based Approach for Misogyny and Sarcasm Detection from Arabic Texts," vol. 2022, 2022.

[32] G. Li, F. Lin, W. Chen, and B. Liu, "Affection Enhanced Relational Graph Attention Network for Sarcasm Detection," *Appl. Sci.*, vol. 12, no. 7, 2022, doi: 10.3390/app12073639.

[33] E. Savini and C. Caragea, "Intermediate-Task Transfer Learning with BERT for Sarcasm Detection," *Mathematics*, vol. 10, no. 5, 2022, doi: 10.3390/math10050844.

[34] V. Govindan and V. Balakrishnan, "A machine learning approach in analysing the effect of hyperboles using negative sentiment tweets for sarcasm detection," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5110–5120, 2022, doi: 10.1016/j.jksuci.2022.01.008.

[35] A. Omar and A. E. Hassanien, "An Optimized Arabic Sarcasm Detection in Tweets using Artificial Neural Networks," in *5th International Conference on Computing and Informatics, ICCI 2022*, 2022, pp. 251–256, doi: 10.1109/ICCI54321.2022.9756102.

[36] R. Kumar and S. Sinha, "Comprehensive Sarcasm Detection using Classification Models and Neural Networks," *8th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2022*, pp. 1309–1314, 2022, doi: 10.1109/ICACCS54159.2022.9785305.

[37] M. S. Razali, A. Abdul Halin, Y. W. Chow, N. Mohd Norowi, and S. Doraisamy, "Context-Driven Satire Detection With Deep Learning," *IEEE Access*, vol. 10, no. August, pp. 78780–78787, 2022, doi: 10.1109/ACCESS.2022.3194119.

[38] A. Kamal and M. Abulaish, "CAT-BiGRU: Convolution and Attention with Bi-Directional Gated Recurrent Unit for Self-Deprecating Sarcasm Detection," *Cognit.*

Comput., vol. 14, no. 1, pp. 91–109, 2022, doi: 10.1007/s12559-021-09821-0.

[39] M. A. Abdelaal, M. A. Fattah, and M. M. Arafa, "Predicting Sarcasm and Polarity in Arabic Text Automatically: Supervised Machine Learning Approach," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 8, pp. 2550–2560, 2022.

[40] P. Kumar and G. Sarin, "WELMSD – word embedding and language model based sarcasm detection," *Online Inf. Rev.*, vol. 46, no. 7, pp. 1242–1256, 2022, doi: 10.1108/OIR-03-2021-0184.

[41] S. Majumdar, D. Datta, A. Deyasi, S. Mukherjee, A. K. Bhattacharjee, and A. Acharya, "Sarcasm Analysis and Mood Retention Using NLP Techniques," *Int. J. Inf. Retr. Res.*, vol. 12, no. 1, pp. 1–23, 2022, doi: 10.4018/ijirr.289952.

[42] P. Kumaran and S. Chitrakala, "A novel mathematical modeling in shift in emotion for gauging the social influential in big data streams with hybrid sarcasm detection," *Concurr. Comput. Pract. Exp.*, vol. 34, no. 3, pp. 1–23, 2022, doi: 10.1002/cpe.6597.

[43] W. Chen, F. Lin, X. Zhang, G. Li, and B. Liu, "Jointly Learning Sentimental Clues and Context Incongruity for Sarcasm Detection," *IEEE Access*, vol. 10, pp. 48292–48300, 2022, doi: 10.1109/ACCESS.2022.3169864.

[44] S. K. Bharti, B. Vachha, R. K. Pradhan, K. S. Babu, and S. K. Jena, "Sarcastic sentiment detection in tweets streamed in real time : a big data approach," *Digit. Commun. Networks*, vol. 2, no. 3, pp. 108–121, 2016, doi: 10.1016/j.dcan.2016.06.002.

[45] A. Syrien, M. Hanumanthappa, and A. Syrien, "Sentiment Polarity through Sarcasm Identification and Detection in Tweets," vol. 13, no. 71, pp. 40794–40801, 2022.

[46] D. K. Sharma, B. Singh, and A. Garg, "An Ensemble Model for detecting Sarcasm on Social Media," *Proc. 2022 9th Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2022*, pp. 743–748, 2022, doi: 10.23919/INDIACom54597.2022.9763115.

[47] K. Kavitha and S. Chittieni, "An Intelligent Metaheuristic Optimization with Deep Convolutional Recurrent Neural Network Enabled Sarcasm Detection and Classification Model," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 2, pp. 304–314, 2022, doi: 10.14569/IJACSA.2022.0130236.

[48] P. Goel, R. Jain, A. Nayyar, S. Singhal, and M. Srivastava, "Sarcasm detection using deep learning and ensemble learning," *Multimed. Tools Appl.*, 2022, doi: 10.1007/s11042-022-12930-z.

[49] D. K. Sharma, B. Singh, S. Agarwal, H. Kim, and R. Sharma, "Sarcasm Detection over Social Media Platforms Using Hybrid Auto-Encoder-Based Model," *Electron.*, vol. 11, no. 18, 2022, doi: 10.3390/electronics11182844.

[50] A. Parkar and R. Bhalla, "Analytical Comparison On Detection Of Sarcasm Using Machine Learning And Deep Learning Techniques," *Int. J. Comput. Digit. Syst.*, vol. 15, no. 1, pp. 1615–1625, 2024, doi: 10.12785/ijcds/1501114.

[51] P. K. M and K. Veeramani, "Harnessing Advanced Learning for Sarcasm Detection,"

2024 Third Int. Conf. Smart Technol. Syst. Next Gener. Comput., pp. 1–5, doi: 10.1109/ICSTSN61422.2024.10671235.

[52] T. Goyal, D. H. Rajeshbai, N. Gopalkrishna, M. Tushar, and H. R. Mamatha, "Mobile Machine Learning Models for Emotion and Sarcasm Detection in Text: A Solution for Alexithymic Individuals," *2024 3rd Int. Conf. Innov. Technol. INOCON 2024*, no. 2018, pp. 1–5, 2024, doi: 10.1109/INOCON60754.2024.10511772.

[53] D. Suhartono, W. Wongso, and A. Tri Handoyo, "IdSarcasm: Benchmarking and Evaluating Language Models for Indonesian Sarcasm Detection," *IEEE Access*, vol. 12, no. June, pp. 87323–87332, 2024, doi: 10.1109/ACCESS.2024.3416955.

[54] K. U. Singh, N. Singh, V. Chaudhary, D. Paliwal, T. Singh, and A. kumar Dewangan, "Enhancing Social Media Sarcasm Detection Using Chicken Swarm Optimization and Graph Neural Networks," *Proc. InC4 2024 - 2024 IEEE Int. Conf. Contemp. Comput. Commun.*, vol. 1, pp. 1–6, 2024, doi: 10.1109/InC460750.2024.10649201.

[55] Z. Karkiner and M. Sert, "Sarcasm Detection in News Headlines with Deep Learning," *32nd IEEE Conf. Signal Process. Commun. Appl. SIU 2024 - Proc.*, pp. 1–4, 2024, doi: 10.1109/SIU61531.2024.10600986.

[56] J. Thambi, S. S. H. Samudrala, S. R. Vadluri, P. C. Nair, and M. Venugopalan, "Sarcasm Detection in News Headlines Using ML and DL Models," *Proc. - 3rd Int. Conf. Adv. Comput. Commun. Appl. Informatics, ACCAI 2024*, no. Ml, pp. 1–8, 2024, doi: 10.1109/ACCAI61061.2024.10601793.

[57] A. Alakrot, A. Dogman, and F. Ammer, "Sarcasm Detection in Libyan Arabic Dialects Using Natural Language Processing Techniques," *2024 IEEE 4th Int. Maghreb Meet. Conf. Sci. Tech. Autom. Control Comput. Eng. MI-STA 2024 - Proceeding*, no. May, pp. 761–767, 2024, doi: 10.1109/MI-STA61267.2024.10599695.

[58] B. Wu, H. Tian, X. Liu, W. Hu, C. Yang, and S. Li, "Sarcasm Detection in Chinese and English Text with Fine-Tuned Large Language Models," *Proc. - 2024 IEEE 10th Conf. Big Data Secur. Cloud, BigDataSecurity 2024*, pp. 47–51, 2024, doi: 10.1109/BigDataSecurity62737.2024.00016.

[59] R. R. Waly, M. Saeed, A. Ashraf, and W. H. Gomaa, "Advancements in Arabic Tweets Sarcasm Detection & Sentiment Analysis with Fine-Tuned Models," *2024 Intell. Methods, Syst. Appl.*, no. Subtask 1, pp. 292–298, 2024, doi: 10.1109/imsa61967.2024.10652758.

[60] S. Puvvada, Kusuma, M. V. Shaik, K. Galla, S. Venkatrama Phani Kumar, and K. Venkata Krishna Kishore, "A Novel Machine Learning Model for Sarcasm Detection on Facebook Comments," *Proc. 3rd Int. Conf. Appl. Artif. Intell. Comput. ICAAIC 2024*, pp. 546–552, 2024, doi: 10.1109/ICAAIC60222.2024.10575012.

[61] M. E. Hassan, M. Hussain, I. Maab, U. Habib, M. A. Khan, and A. Masood, "Detection of Sarcasm in Urdu Tweets Using Deep Learning and Transformer Based Hybrid Approaches," *IEEE Access*, vol. 12, no. May, pp. 61542–61555, 2024, doi: 10.1109/ACCESS.2024.3393856.

[62] J. Dai, “A BERT-Based with Fuzzy logic Sentimental Classifier for Sarcasm Detection,” *2024 7th Int. Conf. Adv. Algorithms Control Eng. ICAACE 2024*, pp. 1275–1280, 2024, doi: 10.1109/ICAACE61206.2024.10548550.

[63] H. Liu, R. Wei, G. Tu, J. Lin, C. Liu, and D. Jiang, “Sarcasm driven by sentiment: A sentiment-aware hierarchical fusion network for multimodal sarcasm detection,” *Inf. Fusion*, vol. 108, no. March, p. 102353, 2024, doi: 10.1016/j.inffus.2024.102353.

[64] Y. Tingre, R. Pingale, N. Choudhary, A. Kale, and N. Hajare, “Sarcasm Detection of Emojis Using Machine Learning Algorithm,” *Proc. InC4 2024 - 2024 IEEE Int. Conf. Contemp. Comput. Commun.*, vol. 1, pp. 1–4, 2024, doi: 10.1109/InC460750.2024.10649041.

[65] B. Sreenivas Prasad, N. Akash Babu, T. Harithikeswar Reddy, A. Vishnu Vardhan Reddy, S. Vekkot, and P. C. Nair, “Sarcasm Detection in Newspaper Headlines,” *2024 4th Int. Conf. Intell. Technol. CONIT 2024*, pp. 1–8, 2024, doi: 10.1109/CONIT61985.2024.10627150.

[66] H. Thakur and J. Srivastava, “Sarcasm Detection with BiLSTM Multihead Attention,” *2024 IEEE 9th Int. Conf. Converg. Technol. I2CT 2024*, pp. 1–7, 2024, doi: 10.1109/I2CT61223.2024.10543816.

[67] Y. Li *et al.*, “An attention-based, context-aware multimodal fusion method for sarcasm detection using inter-modality inconsistency,” *Knowledge-Based Syst.*, vol. 287, no. November 2023, p. 111457, 2024, doi: 10.1016/j.knosys.2024.111457.

[68] J. Pradhan, R. Verma, S. Kumar, and V. Sharma, “An Efficient Sarcasm Detection using Linguistic Features and Ensemble Machine Learning,” *Procedia Comput. Sci.*, vol. 235, pp. 1058–1067, 2024, doi: 10.1016/j.procs.2024.04.100.

[69] Z. Wang, S. jie Hu, and W. dong Liu, “Product feature sentiment analysis based on GRU-CAP considering Chinese sarcasm recognition,” *Expert Syst. Appl.*, vol. 241, no. February 2023, p. 122512, 2024, doi: 10.1016/j.eswa.2023.122512.

[70] V. Grover and H. Banati, “An attention approach to emoji focused sarcasm detection,” *Heliyon*, vol. 10, no. 17, p. e36398, 2024, doi: 10.1016/j.heliyon.2024.e36398.

[71] Q. Abuein, R. M. Al-Khatib, A. Migdady, M. S. Jawarneh, and A. Al-Khateeb, “ArSa-Tweets: A novel Arabic sarcasm detection system based on deep learning model,” *Heliyon*, vol. 10, no. 17, p. e36892, 2024, doi: 10.1016/j.heliyon.2024.e36892.

[72] M. Nirmala, A. H. Gandomi, M. R. Babu, L. D. D. Babu, and R. Patan, “An Emoticon-Based Novel Sarcasm Pattern Detection Strategy to Identify Sarcasm in Microblogging Social Networks,” *IEEE Trans. Comput. Soc. Syst.*, vol. 11, no. 4, pp. 5319–5326, 2024, doi: 10.1109/TCSS.2023.3306908.

[73] M. A. Galal, A. Hassan Yousef, H. H. Zayed, and W. Medhat, “Arabic sarcasm detection: An enhanced fine-tuned language model approach,” *Ain Shams Eng. J.*, vol. 15, no. 6, p. 102736, 2024, doi: 10.1016/j.asej.2024.102736.

[74] P. Tiwari, L. Zhang, Z. Qu, and G. Muhammad, “Quantum Fuzzy Neural Network for multimodal sentiment and sarcasm detection,” *Inf. Fusion*, vol. 103, no. July 2023, p. 102085, 2024, doi: 10.1016/j.inffus.2023.102085.

[75] L. Zhou, X. Xu, and X. Wang, “BNS-Net: A Dual-Channel Sarcasm Detection Method Considering Behavior-Level and Sentence-Level Conflicts,” *2024 Int. Jt. Conf. Neural Networks*, no. C, pp. 1–9, 2024, doi: 10.1109/ijcnn60899.2024.10651549.

[76] J. Wang, Y. Yang, Y. Jiang, M. Ma, Z. Xie, and T. Li, “Cross-modal incongruity aligning and collaborating for multi-modal sarcasm detection,” *Inf. Fusion*, vol. 103, no. August 2023, p. 102132, 2024, doi: 10.1016/j.inffus.2023.102132.

[77] R. A. Fitrianto, A. S. Editya, M. M. H. Alamin, A. L. Pramana, and A. K. Alhaq, “Classification of Indonesian Sarcasm Tweets on X Platform Using Deep Learning,” *Proc. - Int. Conf. Informatics Comput. Sci.*, pp. 388–393, 2024, doi: 10.1109/ICICoS62600.2024.10636904.

[78] Q. Lu, Y. Long, X. Sun, J. Feng, and H. Zhang, “Fact-sentiment incongruity combination network for multimodal sarcasm detection,” *Inf. Fusion*, vol. 104, no. December 2023, p. 102203, 2024, doi: 10.1016/j.inffus.2023.102203.

[79] Z. L. Chia, M. Ptaszynski, M. Karpinska, J. Eronen, and F. Masui, “Initial exploration into sarcasm and irony through machine translation,” *Nat. Lang. Process. J.*, vol. 9, no. September, p. 100106, 2024, doi: 10.1016/j.nlp.2024.100106.

[80] K. Schouten and F. Frasincar, “Finding implicit features in consumer reviews for sentiment analysis,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8541, pp. 130–144, 2014, doi: 10.1007/978-3-319-08245-5_8.

[81] S. S. Eldin, A. Mohammed, A. S. Eldin, and H. Hefny, “An enhanced opinion retrieval approach via implicit feature identification,” *J. Intell. Inf. Syst.*, vol. 57, no. 1, pp. 101–126, 2021, doi: 10.1007/s10844-020-00622-9.

[82] S. I. Omurca and E. Ekinci, “Using Adjusted Laplace Smoothing to Extract Implicit Aspects from Turkish Hotel Reviews,” *2018 IEEE Int. Conf. Innov. Intell. Syst. Appl. INISTA 2018*, 2018, doi: 10.1109/INISTA.2018.8466288.

[83] Z. Liu, Q. Shen, and J. Ma, “Extracting implicit features based on association rules,” *ACM Int. Conf. Proceeding Ser.*, 2018, doi: 10.1145/3265689.3265707.

[84] E. Ekinci and S. İ. Omurca, “Extracting Implicit Aspects based on Latent Dirichlet Allocation,” *Proc. Dr. Consortium–DCAART-(ICAART 2017), Porto, Port.*, no. Dcaart, pp. 17–23, 2017.

[85] L. Zeng and F. Li, “A Classification-Based Approach,” *CCL NLP-NABD 2013*, pp. 190–202, 2013.

[86] R. Abdullah, Suhariyanto, and R. Sarno, “Aspect Based Sentiment Analysis for Explicit and Implicit Aspects in Restaurant Review using Grammatical Rules, Hybrid Approach,

and SentiCircle," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 5, pp. 294–305, 2021, doi: 10.22266/ijies2021.1031.27.

[87] E. H. Hajar and B. Mohammed, "Hybrid approach to extract adjectives for implicit aspect identification in opinion mining," *SITA 2016 - 11th Int. Conf. Intell. Syst. Theor. Appl.*, pp. 1–5, 2016, doi: 10.1109/SITA.2016.7772284.

[88] M. Afzaal, M. Usman, A. C. M. Fong, S. Fong, and Y. Zhuang, "Fuzzy Aspect Based Opinion Classification System for Mining Tourist Reviews," *Adv. Fuzzy Syst.*, vol. 2016, 2016, doi: 10.1155/2016/6965725.

[89] V. Bhatnagar, M. Goyal, and M. A. Hussain, "A proposed framework for improved identification of implicit aspects in tourism domain using supervised learning technique," *ACM Int. Conf. Proceeding Ser.*, vol. 12-13-Augu, pp. 4–7, 2016, doi: 10.1145/2979779.2979835.

[90] Y. Wan, H. Nie, T. Lan, and Z. Wang, "Fine-grained sentiment analysis of online reviews," *2015 12th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2015*, no. 71471019, pp. 1406–1411, 2016, doi: 10.1109/FSKD.2015.7382150.

[91] L. Sun, S. Li, J. Li, and J. Lv, "A novel context-based implicit feature extracting method," *DSAA 2014 - Proc. 2014 IEEE Int. Conf. Data Sci. Adv. Anal.*, pp. 420–424, 2014, doi: 10.1109/DSAA.2014.7058106.

[92] M. Devi Sri Nandhini and G. Pradeep, "A Hybrid Co-occurrence and Ranking-based Approach for Detection of Implicit Aspects in Aspect-Based Sentiment Analysis," *SN Comput. Sci.*, vol. 1, no. 3, pp. 1–9, 2020, doi: 10.1007/s42979-020-00138-7.

[93] T. A. Rana, Y. N. Cheah, and T. Rana, "Multi-level knowledge-based approach for implicit aspect identification," *Appl. Intell.*, vol. 50, no. 12, pp. 4616–4630, 2020, doi: 10.1007/s10489-020-01817-x.

[94] T. A. Rana and Y. N. Cheah, "Hybrid rule-based approach for aspect extraction and categorization from customer reviews," *2015 9th Int. Conf. IT Asia Transform. Big Data into Knowledge, CITA 2015 - Proc.*, 2015, doi: 10.1109/CITA.2015.7349820.

[95] W. Wang, H. Xu, and W. Wan, "Implicit feature identification via hybrid association rule mining," *Expert Syst. Appl.*, vol. 40, no. 9, pp. 3518–3531, 2013, doi: 10.1016/j.eswa.2012.12.060.

[96] S. A, "Implicit Sentiment Identification using Aspect based Opinion Mining," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 3, no. 4, pp. 2184–2188, 2015, doi: 10.17762/ijritcc2321-8169.150491.

[97] H. Xu, F. Zhang, and W. Wang, "Knowledge-Based Systems Implicit feature identification in Chinese reviews using explicit topic mining model," *KNOWLEDGE-BASED Syst.*, vol. 1, no. December, 2014, doi: 10.1016/j.knosys.2014.12.012.

[98] L. Sun and J. Chen, "Joint Topic-Opinion Model For Implicit Feature Extracting," pp. 208–213, 2015, doi: 10.1109/ISKE.2015.17.

[99] L. Liu, Z. Lv, and H. Wang, “Extract Product Features in Chinese Web for Opinion Mining,” vol. 8, no. 3, pp. 627–632, 2013, doi: 10.4304/jsw.8.3.627-632.

[100] Z. Yan, M. Xing, D. Zhang, and B. Ma, “Information & Management EXPRS : An extended pagerank method for product feature extraction from online consumer reviews,” *Inf. Manag.*, vol. 52, no. 7, pp. 850–858, 2015, doi: 10.1016/j.im.2015.02.002.

[101] N. Z. B, A. Selamat, and R. Ibrahim, “Improving Twitter Aspect-Based Sentiment Analysis Using Hybrid Approach,” pp. 151–160, 2016, doi: 10.1007/978-3-662-49381-6.

[102] D. Mining, “Mining explicit and implicit opinions from reviews Farek Lazhar * and Tlili-Guiassa Yamina,” vol. 8, no. 1, pp. 75–92, 2016.

[103] A. Marstawi, N. M. Sharef, and A. Mustapha, “Ontology-based Aspect Extraction for an Improved Sentiment Analysis in Summarization of Product Reviews,” pp. 0–4.

[104] P. Ray and A. Chakrabarti, “A Mixed approach of Deep Learning method and Rule-Based method to improve Aspect Level Sentiment Analysis,” *Appl. Comput. Informatics*, vol. 18, no. 1–2, pp. 163–178, 2022, doi: 10.1016/j.aci.2019.02.002.

[105] O. M. Al-Janabi, M. K. Ibrahim, A. Kanaan-Jebna, O. M. Alyasiri, and H. J. Aleqabie, “An improved Bi-LSTM performance using Dt-WE for implicit aspect extraction,” *2022 Int. Conf. Data Sci. Intell. Comput. ICDSIC 2022*, no. Icdsic, pp. 14–19, 2022, doi: 10.1109/ICDSIC56987.2022.10076109.

[106] Y. Setiowati, A. Djunaidy, and D. O. Siahaan, “Aspect-based Extraction of Implicit Opinions using Opinion Co-occurrence Algorithm,” *2022 5th Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2022*, pp. 781–786, 2022, doi: 10.1109/ISRITI56927.2022.10053003.

[107] C. D. Hoang, Q. V. Dinh, and N. H. Tran, “Aspect-Category-Opinion-Sentiment Extraction Using Generative Transformer Model,” *Proc. - 2022 RIVF Int. Conf. Comput. Commun. Technol. RIVF 2022*, pp. 470–475, 2022, doi: 10.1109/RIVF55975.2022.10013820.

[108] T. Le Thi, T. K. Tran, and T. T. Phan, “Deep Learning Using Context Vectors to Identify Implicit Aspects,” *IEEE Access*, vol. 11, no. March, pp. 39385–39393, 2023, doi: 10.1109/ACCESS.2023.3268243.

[109] N. Dosoula, R. Griep, R. Den Ridder, R. Slangen, K. Schouten, and F. Frasincar, “Detection of multiple implicit features per sentence in consumer review data,” *Commun. Comput. Inf. Sci.*, vol. 615, no. May, pp. 289–303, 2016, doi: 10.1007/978-3-319-40180-5_20.

[110] M. Tubishat and N. Idris, “Explicit and Implicit Aspect Extraction using Whale Optimization Algorithm and Hybrid Approach,” vol. 2, no. IcoIESE 2018, pp. 208–213, 2019, doi: 10.2991/icoiese-18.2019.37.

[111] William and M. L. Khodra, “Generative Opinion Triplet Extraction Using Pretrained Language Model,” *2022 9th Int. Conf. Adv. Informatics Concepts, Theory Appl. ICAICTA*

2022, no. 2020, 2022, doi: 10.1109/ICAICTA56449.2022.9933004.

[112] L. Yu and X. Bai, "Implicit Aspect Extraction from Online Clothing Reviews with Fine-tuning BERT Algorithm," *J. Phys. Conf. Ser.*, vol. 1995, no. 1, 2021, doi: 10.1088/1742-6596/1995/1/012040.

[113] K. Verma and B. Davis, "Implicit Aspect-Based Opinion Mining and Analysis of Airline Industry Based on User-Generated Reviews," *SN Comput. Sci.*, vol. 2, no. 4, 2021, doi: 10.1007/s42979-021-00669-7.

[114] O. M. AL-Janabi, N. H. Ahamed Hassain Malim, and Y. N. Cheah, "Unsupervised model for aspect categorization and implicit aspect extraction," *Knowl. Inf. Syst.*, vol. 64, no. 6, pp. 1625–1651, 2022, doi: 10.1007/s10115-022-01678-5.

[115] A. A. Mar, K. Shirai, and N. Kertkeidkachorn, "Weakly Supervised Learning Approach for Implicit Aspect Extraction †," *Inf.*, vol. 14, no. 11, 2023, doi: 10.3390/info14110612.

[116] H. Benarafa, M. Benkhalifa, and M. Akhloufi, "WordNet Semantic Relations Based Enhancement of KNN Model for Implicit Aspect Identification in Sentiment Analysis," *Int. J. Comput. Intell. Syst.*, vol. 16, no. 1, pp. 1–14, 2023, doi: 10.1007/s44196-022-00164-8.

[117] A. Parkar and R. Bhalla, "A survey paper on the latest techniques for implicit feature extraction using CCC method," *Proc. - 2022 Algorithms, Comput. Math. Conf. ACM 2022*, pp. 20–29, 2022, doi: 10.1109/ACM57404.2022.00012.

LIST OF PUBLICATIONS

JOURNALS

1. Parkar A., Bhalla R., “Analytical comparison on detection of Sarcasm using machine learning and deep learning techniques”, International Journal of Computing and Digital Systems, May 2024
2. Parkar A., Bhalla R., “A Hybrid Framework for Sarcasm Detection Using CB Technique”, International Journal of Basic and Applied Sciences, July 2025

CONFERENCES

1. Parkar A., Bhalla R., “A survey paper on the latest techniques for sarcasm detection using BG method”, 2022 Algorithms, Computing and Mathematics Conference (ACM), August 2022
2. Parkar A., Bhalla R., “A survey paper on the latest techniques for implicit feature extraction using CCC method”, 2022 Algorithms, Computing and Mathematics Conference (ACM), August 2022
3. Parkar A., Bhalla R., “Sentiment analysis on mobile reviews by using machine learning models”, Recent Advances in Computing Sciences (RACS), 2023

Appendix I

Journal

Parkar A., Bhalla R., "Analytical comparison on detection of Sarcasm using machine learning and deep learning techniques", International Journal of Computing and Digital Systems, May 2024

International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (May-24)

http://dx.doi.org/10.12785/ijcds/1501114

Analytical Comparison On Detection Of Sarcasm Using Machine Learning And Deep Learning Techniques

Ameya Parkar[1] and Rajni Bhalla[2]

[1,2]*School of Computer Application,Lovely Professional University, Punjab, India*

Received 8 Jun. 2023, Revised 20 Mar. 2024, Accepted 29 Mar. 2024, Published 1 May. 2024

Abstract: Sentiment Analysis is used in Natural Language processing to detect the opinion of the text/sentence put in by the user. A lot of challenges are faced while detecting the sentiment and one of them is the presence of sarcasm. Sarcasm is very difficult to detect and there could be ambiguity about the presence or absence of sarcasm. Various rule-based methods have been used in the past by researchers to detect sarcasm. However, the results have not been promising. The models developed using machine learning classifiers have gained popularity over the statistical and rule-based methods. Recently, deep learning techniques have been popularly used to detect the presence of sarcasm. In this paper, we have used eight machine language classifiers to detect sarcasm. Deep learning techniques have also being used along with machine learning techniques. An ensemble model has also been trained and tested on both datasets. Bidirectional Encoder Representations from Transformers technique has given the best performance among the deep learning and machine learning techniques with an accuracy score of 92.73% and f-score of 93% on the news headlines dataset and an accuracy score of 75% and f-score of 74% on the Reddit dataset.

Keywords: Sarcasm Detection, Machine Learning (ML), Ensemble model, Deep Learning (DL), Social media

1. Introduction

Social media has become the need of people in their day-to-day lives. People post their opinions, ideas, humor, etc. on social media and share it with other people online. The discussions widely range from sports, politics, movies, etc. and are openly discussed and a lot of information is available online. Websites/Applications such as Twitter allow users to express their own opinions in short text while others such as Reddit, Quora, etc. allow users to express long as well as short opinions. Companies and institutions gather the data relevant to them and try and gauge the public opinion of people about themselves, their products, etc. Sentiment analysis or Opinion mining allows the companies to judge if the people expressing their opinions are talking positively or negatively about them and their products. This helps businesses, organizations, institutes, etc. to understand the sentiment of the people which in turn can lead to promoting and launching a particular product, service, etc. or discarding or making it better. So, sentiment analysis plays an important role and businesses could be putting a lot of effort and money depending on the opinions of the people.

Some users express their sentiments using sarcasm. Sarcasm is the use of text or sentences in which the people mean the opposite of what they want to say. By using sarcasm in their opinions, the polarity of the sentence inverts from positive to negative or vice versa. If the opinions are taken in the form of video then by the gestures of the person and the facial features we can determine if the person is expressing sarcasm or not. If the opinions are taken in the form of audio then by the change of tone we can determine if the person is giving a sarcastic opinion or not. For example, in the context of a cricket game, "Way to go, player" has a very different meaning if said to a player who has got out versus a player who has hit a six. If the player had hit a six, it would be treated as a positive sentiment. However, if the player had got out, it would be a sarcastic opinion. In both cases, judging by the tone and context we can judge if the opinion is sarcastic or not.

If the opinion is only in the form of text then it is very difficult to judge if the opinion is sarcastic or not. In terms of social media such as Twitter, users use hashtags such as #sarcasm, #sarcastic, etc. to denote sarcasm. Some users put emojis such as winking face ;), smiling face, etc. to help the reader understand that the opinion is sarcastic. However, some readers might not understand the emojis and in general might not understand sarcasm. Secondly, the writer might not use the correct hashtags and the correct emojis to express their opinions.

The contributions of this paper are:

1) We collected the dataset from Kaggle. The first dataset is on news headlines and the other dataset is on Reddit posts.

E-mail address: parkar.ameya@gmail.com, rajni.b27@gmail.com

Appendix II

Journal

Parkar A., Bhalla R., “A Hybrid Framework for Sarcasm Detection Using CB Technique”, International Journal of Basic and Applied Sciences, July 2025

Scopus Link of Journal: https://www.scopus.com/sourceid/21101277885

Link of Paper on Journal Website:
https://sciencepubco.com/index.php/IJBAS/article/view/34283/18438

Appendix III

Conference

Parkar A., Bhalla R., "A survey paper on the latest techniques for sarcasm detection using BG method", 2022 Algorithms, Computing and Mathematics Conference (ACM), August 2022

ALGORITHMS, COMPUTING & MATHEMATICS CONFERENCE 2022

ORGANISED BY ADVANCED COMPUTING RESEARCH SOCIETY

This is to certify that the paper titled
"A Survey Paper on the Latest Techniques for Sarcasm Detection using BG Method"
Authored by
Ameya Parkar and Rajni Bhalla
was accepted and presented in
Algorithms, Computing & Mathematics Conference
held on 29 August 2022 at Chennai.
The author
Ameya Parkar, Lovely Professional University, Punjab, India
is appreciated for the contribution and awarded this certificate.

Chennai
29.08.2022

ADVANCED COMPUTING RESEARCH SOCIETY
CHENNAI

Organising Secretary

Appendix IV

Conference

Parkar A., Bhalla R., "A survey paper on the latest techniques for implicit feature extraction using CCC method", 2022 Algorithms, Computing and Mathematics Conference (ACM), August 2022

ALGORITHMS, COMPUTING & MATHEMATICS CONFERENCE 2022

ORGANISED BY ADVANCED COMPUTING RESEARCH SOCIETY

This is to certify that the paper titled
"A survey paper on the Latest Techniques for Implicit Feature Extraction using CCC Method"
Authored by
Ameya Parkar and Rajni Bhalla
was accepted and presented in
Algorithms, Computing & Mathematics Conference
held on 29 August 2022 at Chennai.
The author
Ameya Parkar, Lovely Professional University, Punjab, India
is appreciated for the contribution and awarded this certificate.

Chennai
29.08.2022

ADVANCED COMPUTING RESEARCH SOCIETY
CHENNAI

Organising Secretary

Appendix V

Conference

Parkar A., Bhalla R., “Sentiment analysis on mobile reviews by using machine learning models”, Recent Advances in Computing Sciences (RACS), 2023

LOVELY PROFESSIONAL UNIVERSITY
Transforming Education Transforming India

RACS
SCHOOL OF COMPUTER APPLICATION

Certificate No. 256877

Certificate of Paper Presentation

This is to certify that **Dr./Mr./Ms. Ameya Parkar** of **Lovely Professional University, Punjab** has presented a paper entitled “**Sentiment Analysis on Mobile Reviews by using Machine Learning Models**” in the **1st International Conference on Recent Advances in Computing Sciences (RACS-2022)** held from 4th to 5th November 2022, organized by School of Computer Application at Lovely Professional University, Punjab.

Date of Issue : 30-11-2022
Place : Phagwara (Punjab), India

Prepared by
(Administrative Officer-Records)

Convener
(RACS-2022)

Conference Chair
(RACS-2022)

Session Chair
(RACS-2022)