A project report on

Sentimental Analysis for Walmart Services

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology in Computer Science and Engineering with specialization in Artificial Intelligence and Robotics

By

D L K Trinadh (21BAI1579)

Rishi Patri (21BPS1396)

N Sairam Gopal (21BRS1459)

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

March 2025

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

DECLARATION

I hereby declare that the thesis entitled "**SENTIMENTAL ANALYSIS FOR WALMART SERVICES**" submitted by N SAIRAM GOPAL (21BRS1459), for the award of the degree of Bachelor of Technology in Computer Science and Engineering with specialization in Artificial Intelligence and Robotics, Vellore Institute of Technology, Chennai is a record of bonafide work carried out by me under the supervision of Dr. MARY SHYAMALA L

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Chennai **Signature of the Candidate**

Date:

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled "Sentimental Analysis for Walmart Services" is prepared and submitted by N SAIRAM GOPAL (21BRS1459) to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering with specialization in Artificial Intelligence and Robotics is a bonafide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma, and the same is certified.

Signature of the Guide:
Name:
Dr. Mary Shyamala L
Date:

Signature of the Examiner
Name:
Date:

Signature of the Examiner
Name:
Date:

Approved by the Head of Department,

(Computer Science and Engineering with specialization in Artificial Intelligence and Robotics)

Name: Dr. Harini S

Date:

ABSTRACT

In this project, we present a comprehensive sentiment analysis framework tailored to assess customer perceptions of Walmart's services using real-world review data. Leveraging reviews sourced from Trustpilot through web scraping, we perform extensive preprocessing, including cleaning and tokenization, to prepare the dataset for emotion and sentiment analysis. Utilizing the zero-shot classification capabilities of the pre-trained facebook/bart-large-mnli model, we categorize each review into ten fine-grained emotion labels: happy, joyous, sad, angry, frustrated, excited, disappointed, hopeful, relieved, and surprised. To deepen our analysis, we implement and compare a variety of machine learning and deep learning models—Logistic Regression, CNN, LSTM, RoBERTa, XLNet, and a hybrid model combining Bi-LSTM, BERT, and Attention mechanisms—for their effectiveness in emotion classification. Recognizing that customer feedback often spans multiple service dimensions, we incorporate aspect-based sentiment analysis to identify key focus areas such as Customer Service, Delivery & Orders, In-Store Experience, and Pricing & Billing. Using facebook/bart-large-mnli, we extract aspects and compute confidence scores, followed by sentiment classification through cardiffnlp/twitter-roberta-base-sentiment-latest. This dual-layered approach enables us to determine the sentiment polarity (positive, negative, or neutral) associated with each aspect, offering granular insights into Walmart's service performance from the customer perspective. The results provide valuable implications for customer experience management and service improvement strategies.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr. Mary Shyamala L, Assistant Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Deep Learning.

It is with gratitude that I would like to extend my thanks to the visionary leader Dr. G. Viswanathan our Honorable Chancellor, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan, Dr. G V Selvam Vice Presidents, Dr. Sandhya Pentareddy, Executive Director, Ms. Kadhambari S. Viswanathan, Assistant Vice-President, Dr. V. S. Kanchana Bhaaskaran Vice-Chancellor, Dr. T. Thyagarajan Pro-Vice Chancellor, VIT Chennai and Dr. P. K. Manoharan, Additional Registrar for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dr. Ganesan R, Dean, Dr. Parvathi R, Associate Dean Academics, Dr. Geetha S, Associate Dean Research, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant state, I express ingeniously my whole-hearted thanks to Dr. Harini S, Head of the Department, B.Tech. Computer Science and Engineering with specialization in Artificial Intelligence and Robotics and the Project Coordinators for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staffs at Vellore Institute of Technology, Chennai who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai
Date: **N SAIRAM GOPAL**

CONTENTS

CHAPTER 3

PROPOSED METHODOLOGY

CHAPTER 4

ASPECT EXTRACTION

CHAPTER 5

RESULTS AND ANALYSIS

CHAPTER 6

CONCLUSION AND FUTURE WORK

LIST OF FIGURES

LIST OF ACRONYMS

CNN	Convolutional Neural Network
AI	Artificial Intelligence
XAI	Explainable AI
ML	Machine Learning
DL	Deep Learning
SVM	Support Vector Machine
POS	Parts Of Speech
NLP	Natural Language Processing
BART	Bidirectional and Auto-Regressive Transformers
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
RoBERTa	Robustly Optimized BERT Pretraining Approach
BiLSTM	Bidirectional Long Short-Term Memory
BERT	Bidirectional Encoder Representations from Transformers
AER	Aspect Extraction and Recognition
ABSA	Aspect-Based Sentiment Analysis
LDA	Latent Dirichlet Allocation

Chapter 1

Introduction

1.1 Sentiment Analysis – Background

Sentiment analysis is an effective NLP method utilized for assisting enterprises to understand customers' emotion, opinion, and sentiment. Reviews made by customers play a critical role in the calculation of service quality, responding to pain areas, and succeeding in improving retail organizations. Walmart is the largest chain of retailers receiving thousands of comments on multiple platforms, which helps in understanding diverse aspects of its services. Sentiment analysis of Walmart's services has also been conducted in this study through web scraping of customers' reviews on Trustpilot. The data gathered is tokenized and pre-processed to ensure consistency and uniformity before classification. In Hugging face, we have used a model by Facebook’s BART large MLNI Model, a zero-shot classifier, we classify reviews into ten emotion labels: happy, joyous, sad, angry, frustrated, excited, disappointed, hopeful, relieved, and surprised. Other than that, aspect-based sentiment analysis (ABSA) is performed using CardiffNLP’s Twitter-RoBERTa Model enabling us to find sentiment polarity (positive, negative, or neutral) for key service aspects such as customer service, delivery & orders, in-store experience, and pricing & billing. To improve the accuracy in classification, several machine learning and deep learning models such as logistic regression, CNN, LSTM, RoBERTa, XLNet, and a hybrid model (Bi-LSTM + BERT + Attention Mechanism) are utilized and compared in terms of accuracy, precision, recall, and F1-score. Sarcasm, ambiguous sentences, and class imbalance, which may degrade the performance of classification, are among the most challenging problems for sentiment analysis. Besides, retail sentiment analysis must be domain-adapted to harvest the subtlety of customer interaction. Through aspect-based sentiment classification, this research endeavours to offer concrete recommendations under which Walmart can make optimal service delivery, handle issues with customers effectively, and improve overall customer satisfaction.

1.2 Motivation

The rapid growth of e-business has led to a mushrooming of customer-generated content, including reviews, comments, ratings, and postings on social media. All these vast streams of feedback contain a treasure of information pertaining to likes, dislikes, satisfaction levels, and expectations of the customers. Yet, analyzing so vast an amount of unstructured data is labor-intensive and inefficient to do by hand. Such an issue provides a strong rationale for employing automated techniques to analyze and interpret customers' sentiment quickly and efficiently.

Sentiment analysis fulfils this need by enabling businesses to examine customer opinions and emotions conveyed on various internet sites in a structured way. Customer sentiment is adopted by online businesses in an effort to improve their services and offerings, increase the customer experience, and adjust their marketing strategies based on feedback in real-time. Moreover, the ability to monitor and react to public sentiments helps businesses manage their brand reputation more effectively in a competitive market.

By leveraging the sentiment analysis, the online shopping websites are able to move beyond the traditional feedback system mechanism and gain access to the vast online customer information bank. It not only aids in taking informed, data-driven decisions, but it also results in more cohesive customer relationships and continued business growth.

1.3 Problem Statement

The current age of digitalization witness's customer reviews as the deciding factor in determining business strategies and improving service quality. Walmart, being a large retail chain, receives thousands of reviews on a single day from customers providing their opinions on customer experiences across various service areas such as customer service, delivery & orders, in-store experience, and pricing & billing. However, it is not feasible to analyze these reviews manually due to their vast volume of data. Traditional sentiment analysis methods tend to classify reviews into broad categories such as positive, negative, or neutral and lose the implicit emotional background and specific service information that elicit customers' satisfaction or dissatisfaction. Retail sentiment analysis also comes with challenges such as handling sarcasm, usage of vague language, and imbalance in data among sentiments. In order to overcome such limitations, the present study is seeking to embrace aspect-based sentiment analysis (ABSA) for categorizing Walmart service reviews in a comprehensive sense. Utilizing zero-shot classification with Facebook's BART MNLI Model, we categorize reviews into ten emotion categories in order to study customers' emotions deeper. Further, we seek out crucial aspects of Walmart's services and apply CardiffNLP's Twitter-RoBERTa Model to discover the polarity of the sentiment (positive, negative, or neutral) for each aspect. Besides, some of the most popular machine and deep learning architectures, including logistic regression, CNN, LSTM, RoBERTa, XLNet, and a combination of Bi-LSTM + BERT + Attention mechanism, are employed in an attempt to improve the classification accuracy. The objective of this research is to provide Walmart with a data-driven approach so that customer reviews can be analyzed more efficiently, improving decision-making, improving customer satisfaction, and optimizing services.

1.4 Objectives

The general purpose of applying sentiment analysis in e-commerce services is to systematically determine and analyze the sentiments of the customers for business decision-making and enhancing customer satisfaction. Through the analysis of text on data such as product reviews, social media posts, and customer complaints, the aim is to effectively determine the underlying emotions as negative, positive, or neutral that is employed by customers. This allows e-commerce businesses to make decisions on customers' preference, product quality, and overall market mood.

Besides, the aim entails applying such information in strengthening e-commerce business in various sectors. These include product expansion, customizing markets, enhancing customer support, and reputation management. The second overall aim entails providing real-time monitoring of consumer view, something that companies can react to timely concerning trending activities, potential challenges, or changing consumer behaviors. Lastly, sentiment analysis aims to provide e-commerce sites evidence-based suggestions that translate into increased loyalty, increased sales, and sustainable business expansion.

1.5 Scope of the Project

The project aims to conduct sentiment analysis of Walmart services based on customer reviews gathered from Trustpilot using web scraping. The main aim is to derive meaningful insights from the reviews by labelling the customer sentiments and determining the most prominent factors of Walmart's services that result in customer satisfaction or dissatisfaction. The task employs zero-shot classification with Facebook’s BART MNLI Model to classify reviews into 10 emotion labels, i.e. happy, joyous, sad, angry, frustrated, excited, disappointed, hopeful, relieved, and surprised. Further, ABSA is also carried out using CardiffNLP’s Twitter-RoBERTa-base-sentiment-latest Model for the identification of sentiment polarity (positive, negative, or neutral) for various service aspects like customer service, delivery & orders, in-store experience, and pricing & billing. To improve the performance of classification, the project tests several machine learning and deep learning models such as logistic regression, CNN, LSTM, RoBERTa, XLNet, and a Bi-LSTM + BERT + Attention hybrid model. The models are compared against such critical performance indicators as accuracy, precision, recall, and F1-score in attempting to determine the best method to be used for sentiment classification.

The conclusions of this research will assist Walmart in collecting data-driven insights capturing the opinions of customers so that it can make data-driven decisions, improve its services, and improve customer satisfaction.

This project is useful especially for organizations that need to properly analyze large text data and adopt desired changes in operations. But the work conducted in this paper is text sentiment analysis only. Moreover, even though the models employed are highly efficient, processing sarcasm, ambiguity, and domain-specific subtleties is an area that can be further optimized. Future growth can be in the guise of real-time sentiment monitoring, multi-language sentiment analysis, and cross-integration with other online shopping websites to expand the scope of analysis.

Chapter 2

Background

2.1 Literature Survey

Jabbar et al. [1] present research on real-time sentiment analysis for e-commerce applications, focusing on product reviews from Amazon.com. The study designs and implements a model using a SVM, a machine learning technique, to predict sentiment polarity at both review and sentence levels.

Arora et al. [2] explore the temporal and sentimental dynamics of customer reviews from an Indian e-commerce platform between 2019 and 2023. The study employs NLP techniques with TextBlob and statistical tools like ANOVA to analyze evolving emotions across time. Word cloud visualizations and sentiment classification highlight seasonal variations and feature-specific feedback. While the approach effectively uncovers sentiment trends, its reliance on basic NLP tools may limit accuracy in handling complex linguistic nuances, suggesting the need for more advanced language models in future work.

Sundaram et al. [3] propose an aspect sentiment analysis system for e-commerce using triplet extraction and a BiLSTM-based model. The approach identifies opinion-aspect-sentiment triplets from customer reviews and leverages VADER and syntactic rules for sentiment classification. With an AUC of 0.9 and an average precision of 0.89, the system shows strong performance, including sarcasm detection. However, the reliance on handcrafted preprocessing and basic LSTM layers may limit generalization, highlighting potential for further enhancement through more robust attention-based or generative models.

Wahyudi et al. [4] explore aspect-based sentiment analysis in e-commerce reviews using Latent Dirichlet Allocation (LDA) with Collapsed Gibbs Sampling. The study evaluates sentiment classification accuracy across product categories, comparing general versus category-specific training data. The best performance, with 67.5% accuracy, was achieved using LDA with α=0.001, β=0.001, and 15 topics. Combining general and category-based data slightly improved accuracy by 0.82%. However, the reliance on probabilistic modeling limits linguistic nuance capture, suggesting potential gains through integration with deep learning techniques.

Zulfiker et al.[5] present a machine learning approach for Bangla sentiment analysis on e-commerce reviews. The study constructs a labelled Bangla corpus and evaluates six classifiers using TF-IDF with trigram features. Among them, a hyperparameter-tuned SVM model achieves the highest accuracy of 90.68%. The approach demonstrates strong performance across all metrics. However, the limited dataset size (1,631 reviews) and manual labelling may affect scalability and domain adaptability, suggesting the need for larger, more diverse corpora and automated annotation methods in future work.

Guru Prasad et al. [6] investigate sentiment analysis on e-commerce platforms using transfer learning with transformer-based models. The study applies BERT and its variants for aspect-based sentiment classification, using data from Amazon and Flipkart. The approach demonstrates strong contextual understanding and scalability across different models. However, the large model sizes, computational demands, and challenges in domain adaptation and nuanced aspect detection underscore the trade-offs between performance and efficiency, suggesting the need for careful model selection based on task complexity and resource constraints.

Bharti et al. [7] review efficient machine learning techniques for sentiment analysis of e-commerce customer reviews. The study explores models such as CNN, LSTM, and RMDL, alongside embedding methods like BERT, ELMo, and FastText, to categorize reviews into sentiment classes. It emphasizes the importance of preprocessing and contextual understanding. While the models show promising performance, their effectiveness heavily depends on data quality, feature extraction, and model tuning, suggesting the need for robust hybrid frameworks and domain-specific enhancements.

Chaudari et al. [8] present a sentiment analysis framework for Zomato restaurant reviews using Bi-LSTM and Bi-GRU models. The study combines Word2Vec embeddings with VADER and Sentiment Intensity Analyzer to enhance performance. Experimental results show Bi-GRU outperforms Bi-LSTM in generalization and accuracy, achieving 99.51%. While effective, the Bi-LSTM model showed signs of overfitting, and results varied with dropout rates and batch sizes, highlighting the importance of careful hyperparameter tuning and model selection.

Hakkinen et al. [9] perform sentiment analysis on Tokopedia e-commerce reviews using a lexicon-based method and Naïve Bayes classifier. The study focuses on handling negation, intensifiers, and complex expressions to improve sentiment scoring. With a 400-review dataset, results show 57.5% positive, 22.5% negative, and 20% neutral responses. While the approach is efficient for quick sentiment detection, it may struggle with contextual ambiguity and nuanced expressions, suggesting the need for hybrid or semi-supervised models for better accuracy.

Kathuria et al. [10] analyze sentiment in fashion e-commerce using ML and NLP techniques to study consumer behavior. They apply models like logistic regression, SVM, random forest, and sentiment tools like VADER and TextBlob on reviews and ratings. Logistic regression achieved the highest accuracy (88.18%) for reviews and 80.68% for ratings. While effective, the study highlights challenges in data cleaning, subjectivity handling, and over-attribution of positivity in lexicon-based models, suggesting potential improvements through advanced contextual models.

Al Omari et al. [11] propose a logistic regression-based sentiment classifier for Arabic service reviews in Lebanon. Using TF-IDF features on 3,916 manually collected reviews from Google and Zomato, the model performed well on positive reviews but struggled with negative ones due to data imbalance. Precision for negative sentiment was 0.80, but recall dropped to 0.08, resulting in an ROC AUC of just 0.54. The study highlights challenges in Arabic NLP, particularly sarcasm, code-mixing, and linguistic ambiguity, indicating the need for more balanced datasets and advanced modeling.

Bitto et al. [12] analyze customer sentiment toward Bangladeshi courier services using Bangla NLP and machine learning models. Reviews were scraped from social media for Sundarban, Redx, and Pathao, then classified using six algorithms and N-gram features. Multinomial Naive Bayes with Bigram features achieved the highest accuracy of 90.72%. The study effectively highlights sentiment trends, but limited data size, manual labelling, and underperformance of models like SVM suggest future work with deep learning and larger datasets for enhanced robustness.

Sari et al. [13] apply sentiment analysis to evaluate e-commerce service quality using user reviews from Tokopedia. The study classifies sentiments across five e-Servqual dimensions—reliability, responsiveness, trust, web design, and personalization—using a Naïve Bayes classifier with TF-IDF features. Results show an overall 90% accuracy, with trust and web design receiving the highest positive sentiment. However, personalization was dominated by negative feedback, indicating a need for improved customer engagement. While efficient, the approach is limited by temporal data scope and absence of social media inputs.

Mugil et al. [14] explore a trust-based product analysis model for e-commerce platforms using a hybrid STRUMKNN approach. The model combines sentiment analysis and word separation techniques to assess product reviews and generate trust profiles. By evaluating feedback through polarity-based classification, the system provides product suggestions based on review authenticity and trust weight. Comparative analysis with SVM and K-Means clustering shows that STRUMKNN achieves higher accuracy (92%) in sentiment classification and trust evaluation. While the model demonstrates strong performance, its rule-based sentiment mechanism may limit adaptability to nuanced or sarcastic language, indicating the potential benefit of integrating deep learning-based NLP models in future enhancements.

Panduro et al. [15] investigate the role of sentiment analysis in enhancing product recommendation systems within e-commerce platforms. The study applies various machine learning techniques—such as logistic regression, support vector machines (SVM), and deep learning models like CNN and LSTM—to classify customer review sentiments as positive, negative, or neutral. The methodology includes text preprocessing, feature extraction (e.g., TF-IDF, Word2Vec), and model evaluation using metrics like precision, recall, and F1-score. Experimental results show accuracy levels ranging from 80–90%, with logistic regression performing best. While the approach effectively personalizes recommendations based on user sentiment, challenges remain in handling fake reviews and complex linguistic patterns, suggesting the need for advanced multimodal and ethical AI models in future research.

Taneja et al. [16] explore customer sentiment in Indian fashion e-commerce using a BERT-based deep learning model. Focusing on reviews from platforms like Myntra, Ajio, and Tata Cliq, the study employs BERT's bidirectional contextual embeddings to classify sentiments into positive, neutral, or negative categories. The dataset, comprising over 7,500 reviews, is preprocessed and normalized before classification. The model architecture includes a dropout layer and a fully connected layer atop BERT to predict sentiment classes, achieving an accuracy of 92%, outperforming DistilBERT. While the approach effectively captures nuanced expressions in reviews, occasional misclassifications, especially in neutral cases, highlight challenges posed by subjective human language. Future work aims to incorporate multilingual analysis to cater to India’s diverse linguistic landscape, further improving sentiment classification in regional e-commerce contexts.

Mboungou et al. [17] examine sentiment patterns in customer feedback on a French e-commerce platform, leveraging a dataset of 56,000+ reviews across 16 product categories. The study utilizes advanced machine learning techniques, notably Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models, for emotion classification. Preprocessing includes stop-word removal, lemmatization, and feature extraction using TF-IDF. The system maps sentiments to aspects such as price, quality, shipment, design, and satisfaction. While LSTM aids in capturing long-term dependencies and subtle nuances in French-language reviews, challenges such as sarcasm and ambiguity still require human oversight. Overall, the approach enhances sentiment accuracy and offers actionable insights for businesses aiming to refine product offerings and customer engagement strategies.

Casas-Valadez, et al. [18] conduct a comprehensive bibliometric analysis to investigate the synergy between e-commerce and sentiment analysis in published research from 2007 to 2020. The study utilizes data from Scopus and applies performance metrics and co-occurrence mapping using tools like VOSviewer. It identifies three dominant research clusters: Data Collection & Analysis, Business Intelligence, and Lexicon-based Methods. These themes highlight the evolution of sentiment analysis from basic opinion mining to advanced AI-driven insights.

Jahnavi et al. [19] propose a hybrid sentiment analysis model for social media platforms by combining BERT and RoBERTa architectures. The study utilizes a Kaggle-based labeled dataset enriched with temporal and demographic metadata. Extensive preprocessing, exploratory data analysis, and word cloud visualizations are conducted to uncover textual characteristics. Evaluation with metrics such as accuracy, precision, recall, and F1-score demonstrate the model's superiority over traditional methods like SVM, CNN, and standalone BERT, achieving 82% accuracy.

Samonte et al. [20] explore public sentiment during the COVID-19 pandemic by implementing a hybrid deep learning model that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to analyze Twitter data. The study focuses on capturing spatial features with CNN and temporal dependencies with LSTM to enhance sentiment classification. Tweets were preprocessed through tokenization, stopword removal, and lemmatization, followed by word cloud visualizations and class distribution analysis. The hybrid model outperformed traditional machine learning algorithms in accuracy and contextual understanding. However, its reliance on pre-COVID datasets and general-purpose embeddings may restrict adaptability to evolving slang and domain-specific expressions, indicating the need for dynamic and fine-tuned embedding techniques in future work.

2.2 Research Gaps

Some of the research gaps in the above papers are:

- There is a lack of labelled datasets and pre-trained models for low-resource languages like Bengali, Tamil, and other regional languages in the context of e-commerce sentiment analysis.
- Current deep learning models for sentiment analysis lack interpretability, highlighting a need for integrating XAI techniques to improve trust and transparency.
- Existing models are not sufficiently effective at detecting fake or phoney reviews, particularly against adversarial or bot-generated content, which poses a serious challenge for e-commerce platforms.
- Aspect-Based Sentiment Analysis (ABSA) still struggles with identifying aspect boundaries accurately, especially in complex and domain-specific texts.
- There is a lack of real-time and scalable sentiment analysis systems capable of handling the massive and continuous stream of data generated on large e-commerce platforms.
- Transfer learning models are often too large for practical deployment, indicating a need for research into model compression and adaptation strategies suitable for low-resource or mobile environments.
- Most sentiment analysis systems still use basic classification (positive, negative, neutral), with limited research exploring fine-grained or multi-label sentiment categorization.
- Semi-supervised, few-shot, and active learning approaches are underutilized, despite their potential to address data scarcity issues in e-commerce applications.
- There is insufficient research on longitudinal sentiment tracking to study how customer opinions and satisfaction evolve over time across multiple interactions.
- Cross-platform sentiment integration is still underexplored, and there is a need to fuse sentiment data from e-commerce reviews with social media content to derive richer consumer insights.

Chapter 3

Proposed Methodology

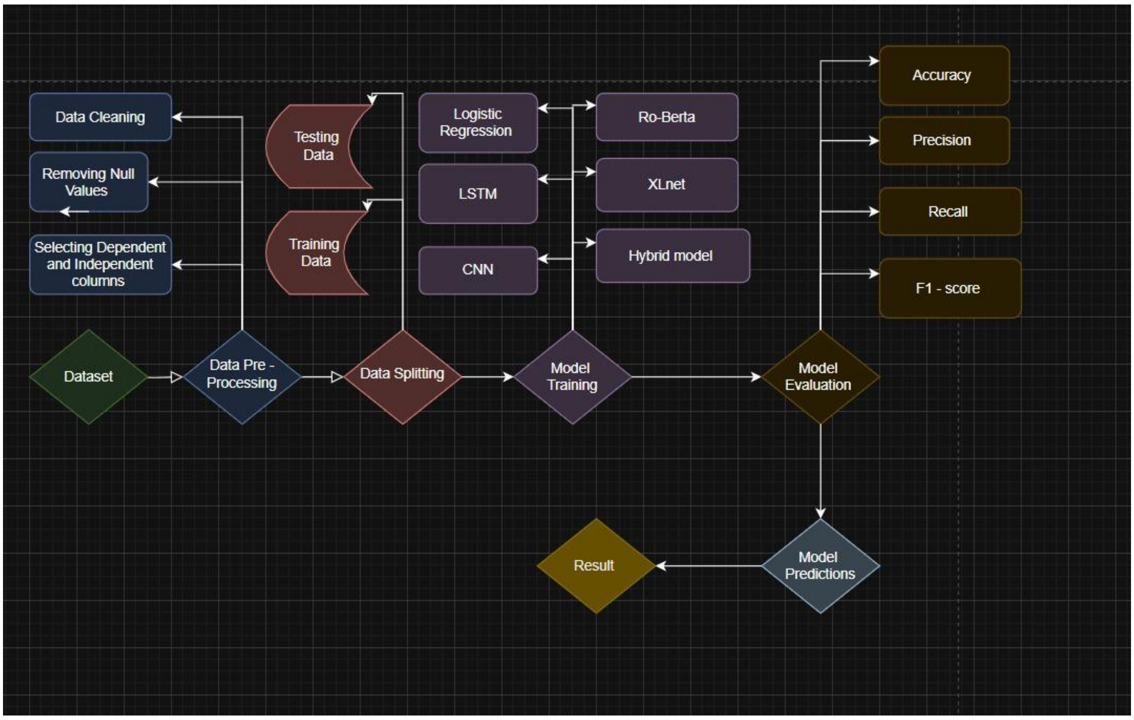


Fig 3.1 Architecture Diagram

3.1 Introduction

One of the most important natural language processing (NLP) methods by which organizations are able to determine what their customers think, feel, and say about their organization is sentiment analysis. Sentiment analysis of Walmart services, in this present study, we will attempt to perform sentiment analysis of Walmart services by sentimentally analyzing the customer reviews gathered from Trustpilot by web scraping. The traditional sentiment analysis categorizes reviews as positive, negative, or neutral but fails to fully understand customers' emotional issues and location-based service issues. To address this limitation, we use zero-shot classification with Facebook/BART-Large-MNLI to classify reviews into ten emotional categories: joyous, happy, sad, angry, frustrated, excited, disappointed, hopeful, relieved, and surprised. We also employ aspect-based sentiment analysis (ABSA) with CardiffNLP/Twitter-RoBERTa-base-sentiment-latest that allows us to measure the sentiment polarity (positive, negative, or neutral) for specified service areas such as customer service, delivery & orders, in-store experience, and pricing & billing. For better sentiment classification accuracy, we leverage multiple machine learning and deep learning models, i.e., logistic regression, CNN, LSTM, RoBERTa, XLNet, and a Bi-LSTM + BERT + Attention mechanism hybrid.

These models help in improved sentiment prediction where customer feedback is labelled appropriately. Watching the trends of the sentiments across various domains, the current work provides valuable information pertaining to customer satisfaction levels, pain areas, and improvement areas. Walmart can possibly employ informed decisions to enhance their service as they try to improve their services and make shopping an improved experience by utilizing such findings. This chapter discusses in detail the data collection, preprocessing, sentiment classification, and aspect-based sentiment analysis techniques used in this research.

3.2 Data Acquisition and Web Scraping

In order to conduct the sentiment analysis of Walmart services, we first collected customer review data from Trustpilot, a well-known review website. Since there was no clear API on Trustpilot to review, we used web scraping techniques in order to get data. Web scraping is programmable text data extraction from the web using BeautifulSoup and Scrapy type tools, which enable us to get structured data effectively.

It was done by discovering Walmart's Walmart review page's HTML structure that is pointing to owners of review text, ratings, timestamps, and customer comments. We created a Python scraping script where reviews would be fetched taking into account support with Walmart's robots.txt file and rate limiting to avoid overloading the server. Secondly, since it was to deal with dynamic load contents, we have used Selenium since it makes web browsers automatable whenever we are dealing with used-content JavaScript loads in scrapes.

After reviewing extracts, raw data was then subjected to initial preprocessing in order to eliminate HTML tags, special characters, duplicate records, and irrelevant information like adverts or metadata. The aggregated dataset provides a good collection of customer feedback towards some aspects of Walmart services like customer service, delivery experience, product quality, and price policy. These data are the basis of our sentiment classification and aspect-based sentiment analysis and allow Walmart to construct a clear view of satisfaction and improvement points for customers.

```python
headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36"
}  # Added headers to prevent request blocking

for i in range(from_page, to_page + 1):
    url = f"https://www.trustpilot.com/review/{company}?page={i}"
    result = requests.get(url, headers=headers)

    if result.status_code != 200:
        print(f"Failed to retrieve page {i}. Status code: {result.status_code}")
        continue  # Skip this page if request fails

    soup = BeautifulSoup(result.content, "html.parser")

    # Extracting Usernames
    soup2list(soup.find_all('span', class_='typography_heading-xs__osRhC'), users)  # Usernames

    # Extracting Total Reviews (Numbers only)
    for user_info in soup.find_all('div', class_='styles_consumerExtraDetails__TylYM'):
        total_reviews_text = user_info.get_text(strip=True)
        total_reviews = ''.join(filter(str.isdigit, total_reviews_text))  # Extract digits only
        userReviewNum.append(total_reviews if total_reviews else "N/A")

    # Extracting Locations (Only country, excluding review count)
    soup2list(soup.find_all('span', class_='typography_body-m__k2UI7 typography_appearance-subtle__PYOVM'), locations)

    # Extracting Dates
    soup2list(soup.find_all('time'), dates, attr='datetime')  # Dates

    # Extracting Ratings
    soup2list(soup.find_all('div', class_='styles_reviewHeader__PuHBd'), ratings, attr='data-service-review-rating')  # Ratings

    # Extracting Review Texts
    for review_content in soup.find_all('div', class_='styles_reviewContent__SCYfD'):
        review_text = review_content.get_text(strip=True)
        reviews.append(review_text if review_text else "N/A")

    sleep(1)  # Prevents request throttling

# Fix: Ensure all lists have the same length
max_length = max(len(users), len(userReviewNum), len(locations), len(dates), len(reviews), len(ratings))

# Pad shorter lists with "N/A"
users += ["N/A"] * (max_length - len(users))
userReviewNum += ["N/A"] * (max_length - len(userReviewNum))
locations += ["N/A"] * (max_length - len(locations))
dates += ["N/A"] * (max_length - len(dates))
reviews += ["N/A"] * (max_length - len(reviews))
ratings += ["N/A"] * (max_length - len(ratings))
```

Fig 3.2 Web Scraping

Username	Date	Review	Rating
Richard	2025-02-14T22:43:05.000Z	Just when you thought Customer Disservice	1
Ronald Mitzel	2025-02-14T06:03:30.000Z	Worst delivery of any store in the United St	1
Armando Garcia Luna	2025-02-12T03:01:21.000Z	Poor customer service & ScamPoor custom	1
Jessica Laurie	2025-02-14T03:06:45.000Z	WalmartWalmart. Where do I even begin?	1
Mike Doyle	2025-02-14T21:37:21.000Z	I bought a cellphone from an OSLI bought a	1
Maggie Barker	2025-02-10T00:34:14.000Z	Walmart doesn't deliverUnfortunately I sigr	1
Michelle Kramer	2025-02-11T14:45:23.000Z	A very unfortunate incident happened toâ€	1
davideagle	2025-02-13T13:10:51.000Z	Went to Walmart on a busy SaturdayWent	3
Joe Moore	2025-02-15T01:45:16.000Z	Walmart could put prices on thereâ€¦Walm	1
Sohan Varma	2025-02-10T00:20:07.000Z	My 70 year old parents ordered mattressâ€	1
Dottie Fisher	2025-02-14T06:25:35.000Z	I LOVE Walmart!I LOVE Walmart!! They tre	5
Shelly C	2025-02-15T03:48:15.000Z	Iâ€™m boycott Walmart because they donâ	1
Troy	2025-02-10T02:12:29.000Z	Low Quality ExperienceThe product quality	1
Curtis Dorman	2025-02-09T18:05:20.000Z	No proper helpSad that every other box cor	1
Scott Hauser	2025-02-10T21:13:13.000Z	Terrific Experience at Walmart Tire DeptWe	1
Michele	2025-02-15T01:50:56.000Z	exchange at customer service with receiptE	5
Jack kaplan	2025-02-08T02:46:10.000Z	It is unbelievable how bad serviceâ€¦It is ur	1
Jesse Brown	2025-02-12T04:28:43.000Z	Employees should be more friendlyI don't o	4
Neal Tanner	2025-02-03T21:23:17.000Z	No wonder Trump wants to make bigâ€¦No	1
Dolores Malish	2025-02-05T06:01:44.000Z	Useless InformationMy bank account was t	1
renee maddox	2025-02-11T16:23:33.000Z	I pay a monthly fee to have myâ€¦I pay a m	2
jay hogan	2025-02-01T16:03:28.000Z	I will never order anything fromâ€¦I will nev	1
Michele Marks	2025-02-04T20:10:53.000Z	DEI stanceSince you wish to roll back your s	1
BlackRock is Evil	2025-01-30T01:31:58.000Z	Walmart criminals cancelled my orderI plac	1
Brianna Long	2025-02-11T15:04:06.000Z	I am writing to express my seriousâ€¦I am v	1

Fig 3.3 Web Scraping

3.3 Data Cleaning and Preprocessing

The second important task is preprocessing and data cleaning to maintain the dataset clean, well-structured, and free of noise so that sentiment analysis will be performed later after Trustpilot scraped Walmart customer reviews. Sentiment classification model accuracy will be disrupted by having irrelevant data, special characters, HTML tags, duplicate records, stopwords, and different formats in raw text data.

The preprocessing pipeline starts with text normalization, where we make all the text lower case so that it is consistent. We remove punctuation, numbers, and special characters so that we can, if possible, remove unwanted noise. We perform tokenization, which is a utility where we divide the text into words or tokens that NLP-based models need. And we also strip away typical stopwords (such as "the," "and," "is") using NLTK or spaCy since they contribute nothing to word meaning of sentiment. Beyond enhancing data quality, we apply lemmatization, which reduces words to their base word (such as "running" to "run") in attempting to keep words in a normalized form and being less redundant. Missing values are also to be addressed; missing or null reviews are dropped or imputed where necessary for analysis. Lastly, pre-processed and cleaned data are held in a formatted state and are available for use to carry out sentiment classification, aspect-based sentiment analysis (ABSA), and machine learning model training.Preprocessing gives us high-quality clean input to our models of sentiment analysis, and that gives us meaningful and accurate customer opinion regarding Walmart's service.

3.4 Tokenization of Reviews

Tokenization is an important natural language processing (NLP) operation that involves splitting text into words or subwords to enable sentiment analysis. We perform tokenization subsequent to cleaning and pre-processing Walmart customer reviews to convert text data into a format that can be used in machine learning as well as deep learning models. In this study, we have utilized two types of tokenization: word tokenization and subword tokenization. Word tokenization splits reviews into words to facilitate word-level sentiment analysis, whereas subword tokenization, for instance, Byte Pair Encoding (BPE), is used in deep learning models such as BERT, RoBERTa, and XLNet, to ensure even out-of-vocabulary words are handled efficiently. We perform tokenization using NLTK, spaCy, and Hugging Face's tokenizer, which convert text into numerical tokens (token IDs) that must be input into models while training. Tokenized reviews serve as the foundation for zero-shot classification using Facebook/BART-Large-MNLI and aspect-based sentiment analysis (ABSA) using CardiffNLP/Twitter-RoBERTa-base-sentiment-latest, in order to determine accurate sentiment classification and aspect extraction.

```
[8] data.head()
```

	Username	Total reviews	Date	Review	Rating	Cleaned_Review	Tokenized_Review	Emotion
0	Richard	37K reviews	2025-02-14T22:43:05.000Z	Just when you thought Customer Disservice coul...	1	just when you thought customer disservice coul...	[101, 2074, 2043, 2017, 2245, 8013, 4487, 1811...	sad
1	Ronald Mitzel	3.6K reviews	2025-02-14T06:03:30.000Z	Worst delivery of any store in the United Stat...	1	worst delivery any store united statesi paid m...	[101, 5409, 6959, 2151, 3573, 2142, 2163, 2072...	frustrated
2	Armando Garcia Luna	14K reviews	2025-02-12T03:01:21.000Z	Poor customer service & ScamPoor customer serv...	1	poor customer service scampoor customer servic...	[101, 3532, 8013, 2326, 8040, 16613, 16506, 80...	disappointed
3	Jessica Laurie	3reviews	2025-02-14T03:06:45.000Z	WalmartWalmart. Where do I even begin? It's ...	1	walmartwalmart where do i even begin its like ...	[101, 24547, 22345, 13476, 22345, 2073, 2079, ...	frustrated
4	Mike Doyle	1review	2025-02-14T21:37:21.000Z	I bought a cellphone from an OSLI bought a cel...	1	i bought cellphone from an osli bought cellpho...	[101, 1045, 4149, 3526, 9864, 2013, 2019, 9808...	disappointed

Fig 3.4 Tokenization of Reviews

3.5 Emotion Labeling and Classification

Emotion labelling is an important sentiment analysis task that assists in understanding the emotional tone of customer reviews outside the ordinary positive, negative, or neutral label. In this research, we categorize Walmart customer reviews into ten emotion labels: happy, joyous, sad, angry, frustrated, excited, disappointed, hopeful, relieved, and surprised. This categorization facilitates more detailed sentiment analysis, which assists Walmart in ascertaining the emotional reaction of customers towards various facets of its services.

In order to do this, we employ Facebook/BART-Large-MNLI, a zero-shot pre-trained classifier, outputting confidence scores on all emotional labels as a function of the review's sentiment probability expressing such a sentiment. This enables sensitive emotional nuance to be captured, e.g., frustrating versus disappointing and exciting versus joyful.

With the assistance of emotion labelling, we have an increased understanding of feelings among customers, enabling Walmart to capture repetitive emotional patterns in consumer complaints. Having scrutinized this data, Walmart can improve quality of service, personalize customer experience, and better handle dissatisfaction. The labelled emotions are also an input for aspect-based sentiment analysis (ABSA), which enables Walmart to identify what service features elicit a certain type of emotional response.

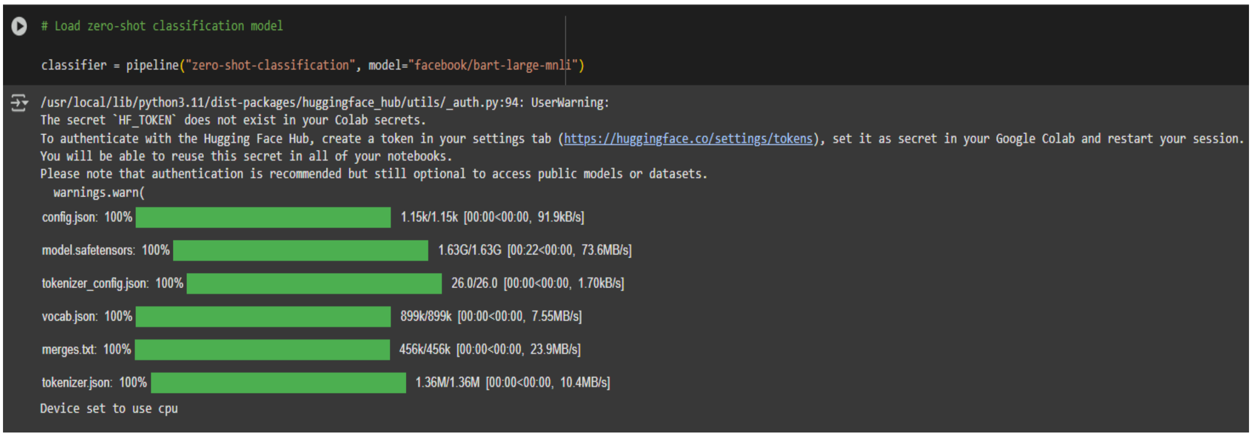


Fig 3.5 Zero-shot classification using Facebook/BART-Large-MNLI

3.6 Machine Learning Models for Sentiment Classification

1. Logistic Regression

Logistic Regression is a popular machine learning algorithm for binary and multi-class classification problems, including sentiment analysis. It is a linear model that estimates the probability of a binary outcome using a logistic function on a weighted sum of input features. In sentiment analysis, Logistic Regression is used to classify text (e.g., Walmart reviews) into sentiment classes, e.g., positive, negative, and neutral.

Logistic Regression achieves this through the initial conversion of the text data into numerical features that can be input into the model. Typically, methods used for extracting features from text include TF-IDF (Term Frequency-Inverse Document Frequency), Bag of Words, or word embeddings. These features extract information on the frequency and significance of words or terms in the text so that the model can recognize key information on sentiment.

After the features have been extracted, Logistic Regression assigns each feature a weight depending on how much it contributes to the overall sentiment. The weights are then used by the model to compute a score, and the score is run through a sigmoid function to produce a prediction for the probability of a certain sentiment (e.g., negative or positive).

The output is a binary output, or for an extended model, a multi-class output for more than a single sentiment class.

The biggest benefits of employing Logistic Regression for sentiment analysis are that it is simple, comprehensible, and comparatively low in computational cost. It is simple to comprehend how it does its predictions, and this can be a humongous advantage when needing to explain model outcomes to decision-makers. Moreover, Logistic Regression learns rapidly, even for big data, so it is an excellent option for use in applications where computational efficiency matters.

2. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are deep neural networks that have become increasingly popular for application in image processing but also in natural language processing (NLP) applications like sentiment analysis. CNNs are employed in sentiment analysis to detect neighbourhood patterns and co-occurrences of text via convolutional filters on the input and, in this case study, customer review text.

The convolutional layers work by scanning the input text with filters and producing feature maps to identify significant patterns in the review. The feature maps are then passed through a max-pooling layer, which reduces the feature maps by taking the most significant information, e.g., the most indicative words or phrases of sentiment. The output of the convolutional layers is flattened and passed to fully connected layers to produce a final classification output, in which the review is mapped to a sentiment label (positive, negative, neutral).

One of the strongest advantages of sentiment analysis with CNNs is that they can automatically learn and obtain good features from the text, without tedious manual feature engineering typically required with other methods. This is especially valuable for sentiment analysis, because context and local patterns inside sentence play such a big role. CNNs are engineered to capture useful local dependencies, like the words or n-grams that convey sentiment and affective tone.

But CNNs are not without their limitations. They are very good at capturing neighbourhood patterns, perhaps not quite so good at capturing long-distance dependencies or context which is about noticing the relationship between distant words within the sentence. This is a very critical problem with sentiment analysis since the overall sentiment of a review can be based on words that are dispersed all over the text, not consecutive words.

In general, CNNs are an inexpensive computational but efficient model to apply on sentiment analysis tasks, especially when computation is inexpensive or the task is defined as local texture recognition or word dependencies in restricted windows. They appear to have robust performance-efficiency ratios and hence are an attractive solution to sentiment analysis for real-time.

3. Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Networks (RNNs) which are specifically developed to address problems of typical RNNs, especially long-term dependencies and vanishing gradients. LSTMs perform very well for sequential inputs such as text in which the context of a sentence is based upon words that occur much earlier or further.

In sentiment analysis, e.g., Walmart review analysis, LSTMs operate in a manner where they first transform the words into word embeddings (e.g., GloVe or BERT) and then input them into a sequence of LSTM units. Each unit has input gates, forget gates, and output gates, which assist the model in deciding what to remember or forget as it traverses the sequence.

LSTMs are particularly effective at recognizing context and are particularly beneficial when processing mixed or compound emotions within long sentences. For example, they can recognize positive and negative emotions in a review such as: "The product was amazing, but the customer service was poor."

While they are excellent at identifying weak and distant sentiment patterns like sarcasm or negation, LSTMs have their downsides. They use a lot of training data, are computationally expensive, and are vulnerable to overfitting when handling small data. Regardless of that, with the help of the likes of Bi-LSTM and attention, LSTMs are still a handy tool for deep sentiment mining of text data.

3.7 Transformer-Based Deep Learning Models

1. RoBERTa

RoBERTa, (Robustly Optimized BERT Pretraining Approach), is a transformer-based model, a variant of the Bidirectional Encoder Representations from Transformers (BERT) which tries to outperform a wide range of natural language processing (NLP) tasks, such as sentiment analysis. Whereas BERT is optimized by losing the Next Sentence Prediction (NSP) task and training on more mini-batches and longer sequences.

RoBERTa applies a transformer model founded on self-attention mechanisms in an effort to understand the context of every word within a sentence with respect to all the other words within the sentence, both preceding it and following it. Two-way context enables the model to distinguish better between the words so it can understand better the underlying sentiment, such as sarcasm, irony, or complex emotions, that other models struggle to understand.

One of the advantages of RoBERTa over standard models such as Logistic Regression or CNNs is that it is able to learn deep word dependencies and also extract long-range dependencies in a sentence. This is helpful particularly while doing sentiment analysis, where the sentiment is not only dependent on the words it is in proximity with but also on the quantity of context in the direction of the sentence or even sentences. RoBERTa's bi-directional attention also allows it to learn long-distance dependencies such that it can efficiently classify sentiment in reviews containing subtle or complex emotional expressions.

Moreover, RoBERTa is also a pre-trained model, i.e., it was previously pre-trained on huge volumes of text data, and therefore can generalize quite well to other sentiment analysis tasks even if only given very little task-specific training data. This pretraining removes from the requirement of large, labeled datasets on which to train the model anew, which is normally one of the largest bottlenecks of NLP tasks.

RoBERTa is in general an excellent sentiment analysis solution, particularly while handling large and complex data and where derivation of implicit contextual sentiment is a must. Its capability to learn deep relationships in the text and more than BERT, it's a value addition worth investigating while analyzing customer review, customer sentiment, and inferring key emotional drivers from product and service feedback.

2. XLNet for Text Analysis

XLNet is a transformer model developed by Google that is better than models like BERT and RoBERTa by incorporating the strengths of both autoregressive and autoencoding approaches. Unlike BERT, which masks words during training, XLNet uses permutation-based training—allowing it to learn context from any word order without masking. This enables it to learn more accurately about the detailed relationships and dependencies between words, which is very useful for sentiment analysis.

In sentiment analysis tasks, XLNet processes customer reviews by converting them into embeddings and applying self-attention to generate contextual representations. These are then used to classify the sentiment as positive, negative, or neutral. Its ability to model word interactions better and more flexibly makes it most appropriate for detecting subtle or blended sentiments in text.

Another important strength of XLNet is how transferable it is—it performs extremely well on new tasks with minimal fine-tuning because it was pre-trained on massive datasets. However, as with all big transformer models, training it is computationally heavy, requiring top-shelf hardware and large training times. It also overfits to small datasets.

Despite these limitations, XLNet achieves state-of-the-art performance in sentiment analysis by heavily modeling language and picking up nuanced emotional signals in text and therefore is a suitable candidate to analyze complex customer feedback like Walmart reviews.

3. Hybrid Model

The Bi-LSTM + BERT + Attention model is an integration of the strengths of three state-of-the-art techniques to strengthen the performance of sentiment analysis. Bi-LSTM captures contextual meaning in both directions of a sentence, which is effective in capturing remote word relationships. BERT, a transformer model, also strengthens the understanding by studying the entire sentence context bidirectionally, which is very helpful in explaining complex or shifting sentiments. The Attention Mechanism enhances this setup by giving more weight to affectively pertinent words—like "excellent" or "terrible"—in the ultimate sentiment prediction.

Through this amalgamation of both elements, the model is sufficiently capable of handling mixed-sentiment and subtle reviews. And for that, this accuracy boost has a cost: the model is computationally heavy and requires longer periods and more resources of training than in lightweight models. Still, this hybrid model structure works wonders to tap into more meaningful customer feedback insights.

Chapter 4

Aspect Extraction

4.1 Introduction

Sentiment classification is an important task in customer feedback analysis, as it enables companies to know how their services are viewed by users. In this research, we seek to categorize Walmart customer reviews into insightful sentiment categories using zero-shot classification and aspect-based sentiment analysis (ABSA). While conventional sentiment classification merely classifies reviews as positive, negative, or neutral, our method delves further by recognizing particular emotions and examining sentiment polarity for various aspects of Walmart's services.

For this purpose, we utilize CardiffNLP’s Twitter RoBERTa-base-sentiment-latest Model, where we label every aspect as Customer Service, In-Store Experience, Delivery & Orders, and Pricing & Billing.

This approach provides Walmart a more detailed understanding of customer attitudes, which can be used to ascertain on which aspects of their services the most positive or negative feedback is received. The information gained through sentiment categorization can be used to improve customer service, delivery performance, in-store experience, and prices, leading to overall increased customer satisfaction.

4.2 Zero-Shot Classification

Zero-shot classification is an effective NLP technique that makes it possible for text to be labelled into predetermined labels without relying on a human-labelled training set. Within this paper, we employ CardiffNLP’s Twitter RoBERTa-base-sentiment-latest Model, a state-of-the-art zero-shot classification model, for labelling Walmart customers' reviews under ten emotional categories: happy, joyous, sad, angry, frustrated, excited, disappointed, hopeful, relieved, and surprised.This approach enables deeper insight into customer emotions beyond the typical sentiment analysis, which typically detects only positive, negative, or neutral sentiments. The model treats each review as input and provides confidence scores to different emotion labels as a function of how likely it is that the review holds that sentiment. Since zero-shot classification doesn't require training in domains, it supports the sentiment analysis for big datasets as quickly and scale-up. It also allows detecting subtle emotional subtleties from customers' reviews to make insights more beneficial to Walmart.

By applying CardiffNLP’s Twitter RoBERTa-base-sentiment-latest Model, we can accurately classify customer reviews so that Walmart is able to better find patterns of customer sentiment and resolve cases with customer service, delivery, in-store experience, and prices. The output of this classification is then applied to aspect-based sentiment analysis (ABSA) to provide a comprehensive evaluation of Walmart's quality of service.

4.3 Sentiment Analysis using CardiffNLP/ Twitter-RoBERTa

CardiffNLP’s Twitter RoBERTa-base-sentiment-latest Model is a pre-trained transformer-based model fine-tuned specifically for sentiment analysis on social media data, specifically Twitter. Built on the RoBERTa architecture (which builds on BERT by removing the Next Sentence Prediction task and training on more data), this model is tuned for the informal, frequently abbreviated, and slang-laden language one might expect to encounter on sites like Twitter. As Walmart reviews can include colloquial wordings, jargon, or even emojis within them. The model is able to classify text into finer sentiment categories like "very positive" or "very negative," or more general sentiment categories like positive, negative, and neutral. It also manages the nuances of emotion in the guise of sarcasm, slang, and colloquial language very well and is thus best placed to examine user-generated content like Walmart reviews. Also, as it has been trained on a humongous dataset gathered from social media, the model is adept at dealing with diverse expressions and words to ensure even brief or subtle customer sentiments are properly reflected. It allows improved and context-based sentiment labelling of reviews, and it enhances the overall sentiment analysis process for Walmart's services.

```python
# Load the Extracted Aspects Dataset

file_path = "/kaggle/input/extracted-aspects/aspect_extracted_reviews.csv"
df = pd.read_csv(file_path)
```
Python

```python
# Load Improved RoBERTa Sentiment Model

sentiment_analyzer = pipeline("sentiment-analysis", model="cardiffnlp/twitter-roberta-base-sentiment-latest", device=0)  # Use latest model

tokenizer = AutoTokenizer.from_pretrained("cardiffnlp/twitter-roberta-base-sentiment-latest")
```
Python

```
Some weights of the model checkpoint at cardiffnlp/twitter-roberta-base-sentiment-latest were not used when initializing RobertaForSequenceClassification: ['roberta.pooler.dense.bias', 'roberta.pooler.dense.weight']
- This IS expected if you are initializing RobertaForSequenceClassification from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a
- This IS NOT expected if you are initializing RobertaForSequenceClassification from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenc
Device set to use cuda:0
```

Fig 4.1 Sentiment Analysis using CardiffNLP/ Twitter-RoBERTa

4.4 Aspect Extraction Using NLP

Aspect extraction is another important process in sentiment analysis that assists in uncovering specific topics or service-related aspects being commented on in customer reviews. In our research, we extract aspects like Customer Service, Delivery & Orders, In-Store Experience, and Pricing & Billing from Walmart customers' reviews to understand more about the sentiments of customers for every service category. Conventional sentiment analysis gives a general positive, negative, or neutral label, but aspect-based sentiment analysis (ABSA) enables us to identify sentiment polarity for every particular aspect discussed in a review. In order to obtain aspect extraction, we use zero-shot classification using CardiffNLP Model, which helps us classify text into pre-defined categories without needing a manually annotated dataset.

This model provides confidence scores to various aspects depending on how likely a review is to fall into a specific category. Utilizing natural language processing (NLP) methods like named entity recognition (NER), dependency parsing, and keyword-based extraction, we are able to accurately extract relevant aspects from unstructured text. The features are subsequently employed in the subsequent analysis phase, where we calculate the sentiment polarity (positive, negative, or neutral) employing this CardiffNLP Modlel. This allows for an in-depth analysis of Walmart's service performance such that areas of strength and weakness can be ascertained from customers' feedback.

	A	B
1	review	predicted_aspects
2	Just when you thought Customer Disservice couldn't I	Customer Service, In-Store Experience, Delivery & Orders
3	Worst delivery of any store in the United StatesI paid	Customer Service, Delivery & Orders, In-Store Experience
4	Poor customer service & ScamPoor customer service,	Customer Service, Delivery & Orders, Pricing & Billing
5	WalmartWalmart. Where do I even begin? It's like a	In-Store Experience, Pricing & Billing, Customer Service
6	I bought a cellphone from an OSLI bought a cellphone	Customer Service, Pricing & Billing, In-Store Experience
7	Walmart doesn't deliverUnfortunately I signed up for	Customer Service, Delivery & Orders, In-Store Experience
8	A very unfortunate incident happened toâ€¦A very un	In-Store Experience, Customer Service, Pricing & Billing
9	Went to Walmart on a busy SaturdayWent to Walma	In-Store Experience, Customer Service, Pricing & Billing
10	Walmart could put prices on thereâ€¦Walmart could	Customer Service, Pricing & Billing, In-Store Experience
11	My 70 year old parents ordered mattressâ€¦My 70 ye	Customer Service, Delivery & Orders, Pricing & Billing
12	I LOVE Walmart!I LOVE Walmart!! They treat my like	Customer Service, Delivery & Orders, In-Store Experience
13	Iâ€™m boycott Walmart because they donâ€™tâ€¦Iâ	Customer Service, In-Store Experience, Pricing & Billing
14	Low Quality ExperienceThe product quality of their o	In-Store Experience, Customer Service, Pricing & Billing
15	No proper helpSad that every other box comes dama	Customer Service, In-Store Experience, Pricing & Billing
16	Terrific Experience at Walmart Tire DeptWe arrived a	Customer Service, In-Store Experience, Delivery & Orders
17	exchange at customer service with receiptExchange a	Customer Service, In-Store Experience, Delivery & Orders
18	It is unbelievable how bad serviceâ€¦It is unbelievabl	Customer Service, Delivery & Orders, Pricing & Billing
19	Employees should be more friendlyI don't often go to	In-Store Experience, Customer Service, Pricing & Billing
20	No wonder Trump wants to make bigâ€¦No wonder T	Delivery & Orders, Customer Service, Pricing & Billing
21	Useless InformationMy bank account was the object	Customer Service, Pricing & Billing, Delivery & Orders
22	I pay a monthly fee to have myâ€¦I pay a monthly fee	Delivery & Orders, Customer Service, Pricing & Billing
23	I will never order anything fromâ€¦I will never order a	Customer Service, Pricing & Billing, Delivery & Orders
24	DEI stanceSince you wish to roll back your stance on	In-Store Experience, Delivery & Orders, Customer Service
25	Walmart criminals cancelled my orderI placed an ord	Customer Service, Delivery & Orders, In-Store Experience
26	I am writing to express my seriousâ€¦I am writing to	In-Store Experience, Customer Service, Pricing & Billing

Fig 4.2 Aspect Extraction

4.5 Assigning Sentiment Scores to Aspects

Having pulled out head terms like Customer Service, Delivery & Orders, In-Store Experience, and Pricing & Billing from Walmart reviews, it is now time to overlay sentiment scores on each area to measure how customers feel about some aspects of the service. It gives a more specific, more accurate image of customer opinion. The method starts with aspect extraction where NLP methods such as keyword extraction or named entity recognition (NER) identify the involved aspects in every review. After identification of aspects, the text for every aspect is isolated and separated for sentiment independently. For this, the CardiffNLP’s Twitter-RoBERTa Model is employed on every piece of text, and it labels each section of the review as either positive, negative, or neutral. The sentiment can be divided into even more levels in some cases, like "very positive," "positive," "neutral," "negative," or "very negative," depending on the capacity of the model. The sentiment score per category is subsequently summed for all reviews and a total sentiment score is returned for each area of service. The aspect level sentiment analysis provides a more detailed view of customers' remarks to reveal true positives and negatives about Walmart services. For example, if customer service is constantly scored as negative, this might point to a specific area that requires improvement, while positive delivery sentiment would mean that this aspect of Walmart's service is fine. Putting a sentiment score on each part, analysis yields actionable results which can be used for driving enhancement in certain areas that result in a better customer experience.

4.6 Confidence Score Calculation for Aspects

Following the extraction of aspects from customer reviews at Walmart, the following step is to compute confidence scores to determine to what extent a review pertains to a specific aspect. Confidence scores are the chances of a review belonging to a certain category with higher confidence scores indicating stronger correlation between reviews and aspects. This allows for more accurate aspect-based sentiment analysis. In this research, we employ CardiffNLP Model, a zero-shot classification model, to provide confidence scores for dimensions like Customer Service, Delivery & Orders, In-Store Experience, and Pricing & Billing. The model goes through every review and gives a probability score to every predefined feature, which shows how much the feature is related to the review. The higher the confidence scores, the higher the correlation between the review and the respective feature. Such filtering removes reviews that can talk about many things but have low correlation with the prominent features in question.

Through the use of confidence scores, we include only highly relevant reviews in aspect-based sentiment analysis (ABSA). This enhances the precision of sentiment classification with CardiffNLP Model, which classifies each aspect-specific review as positive, negative, or neutral. With confidence scores, the use of a more data-driven methodology is ensured, providing dependable and actionable sentiment insights to help enhance Walmart's services.

```python
# Show Sample Results

print(aspect_df.head())
```

```
                                              review  \
0  Just when you thought Customer Disservice coul...
1  Worst delivery of any store in the United Stat...
2  Poor customer service & ScamPoor customer serv...
3  WalmartWalmart.  Where do I even begin?  It's ...
4  I bought a cellphone from an OSLI bought a cel...

                                   predicted_aspects confidence_scores
0  Customer Service, In-Store Experience, Deliver...   0.87, 0.2, 0.15
1  Customer Service, Delivery & Orders, In-Store ...  0.98, 0.77, 0.21
2  Customer Service, Delivery & Orders, Pricing &...  0.89, 0.87, 0.63
3  In-Store Experience, Pricing & Billing, Custom...   0.1, 0.04, 0.03
4  Customer Service, Pricing & Billing, In-Store ...  0.58, 0.16, 0.16
```

	A	B	C	D	E	F	G	H	I	J	K	L
1	review	predicted_aspects	confidence_scores	aspect_sentiments								
2	Just when you thought (	Customer Service,	0.87, 0.2, 0.15	{'Customer Service': 'negative', 'In-Store Experience': 'negative', 'Delivery & Orders': 'negative'}								
3	Worst delivery of any st	Customer Service,	0.98, 0.77, 0.21	{'Customer Service': 'negative', 'Delivery & Orders': 'negative', 'In-Store Experience': 'negative'}								
4	Poor customer service &	Customer Service,	0.89, 0.87, 0.63	{'Customer Service': 'negative', 'Delivery & Orders': 'negative', 'Pricing & Billing': 'negative'}								
5	WalmartWalmart. Whe	In-Store Experienc	0.1, 0.04, 0.03	{'In-Store Experience': 'negative', 'Pricing & Billing': 'negative', 'Customer Service': 'negative'}								
6	I bought a cellphone frc	Customer Service,	0.58, 0.16, 0.16	{'Customer Service': 'negative', 'Pricing & Billing': 'negative', 'In-Store Experience': 'negative'}								
7	Walmart doesn't delive	Customer Service,	0.94, 0.86, 0.3	{'Customer Service': 'negative', 'Delivery & Orders': 'negative', 'In-Store Experience': 'negative'}								
8	A very unfortunate incic	In-Store Experienc	0.83, 0.71, 0.3	{'In-Store Experience': 'negative', 'Customer Service': 'negative', 'Pricing & Billing': 'negative'}								
9	Went to Walmart on a l	In-Store Experienc	0.98, 0.93, 0.27	{'In-Store Experience': 'negative', 'Customer Service': 'negative', 'Pricing & Billing': 'negative'}								
10	Walmart could put price	Customer Service,	0.99, 0.97, 0.93	{'Customer Service': 'negative', 'Pricing & Billing': 'negative', 'In-Store Experience': 'negative'}								
11	My 70 year old parents	Customer Service,	0.97, 0.72, 0.38	{'Customer Service': 'negative', 'Delivery & Orders': 'negative', 'Pricing & Billing': 'negative'}								
12	I LOVE Walmart!I LOVE	Customer Service,	1.0, 0.86, 0.55	{'Customer Service': 'positive', 'Delivery & Orders': 'positive', 'In-Store Experience': 'positive'}								
13	Iâ€™m boycott Walmar	Customer Service,	0.24, 0.1, 0.09	{'Customer Service': 'negative', 'In-Store Experience': 'negative', 'Pricing & Billing': 'negative'}								
14	Low Quality Experience	In-Store Experienc	0.83, 0.0, 0.0	{'In-Store Experience': 'negative', 'Customer Service': 'negative', 'Pricing & Billing': 'negative'}								
15	No proper helpSad that	Customer Service,	0.95, 0.19, 0.12	{'Customer Service': 'negative', 'In-Store Experience': 'negative', 'Pricing & Billing': 'negative'}								
16	Terrific Experience at W	Customer Service,	0.72, 0.6, 0.07	{'Customer Service': 'neutral', 'In-Store Experience': 'neutral', 'Delivery & Orders': 'neutral'}								
17	exchange at customer s	Customer Service,	0.98, 0.85, 0.43	{'Customer Service': 'positive', 'In-Store Experience': 'positive', 'Delivery & Orders': 'positive'}								
18	It is unbelievable how b	Customer Service,	0.99, 0.84, 0.49	{'Customer Service': 'negative', 'Delivery & Orders': 'negative', 'Pricing & Billing': 'negative'}								
19	Employees should be m	In-Store Experienc	0.75, 0.54, 0.31	{'In-Store Experience': 'positive', 'Customer Service': 'positive', 'Pricing & Billing': 'positive'}								
20	No wonder Trump want	Delivery & Orders,	0.91, 0.9, 0.25	{'Delivery & Orders': 'negative', 'Customer Service': 'negative', 'Pricing & Billing': 'negative'}								
21	Useless InformationMy	Customer Service,	0.78, 0.26, 0.21	{'Customer Service': 'negative', 'Pricing & Billing': 'negative', 'Delivery & Orders': 'negative'}								
22	I pay a monthly fee to h	Delivery & Orders,	0.85, 0.59, 0.24	{'Delivery & Orders': 'negative', 'Customer Service': 'negative', 'Pricing & Billing': 'negative'}								
23	I will never order anythi	Customer Service,	0.82, 0.31, 0.1	{'Customer Service': 'negative', 'Pricing & Billing': 'negative', 'Delivery & Orders': 'negative'}								
24	DEI stanceSince you wis	In-Store Experienc	0.47, 0.04, 0.02	{'In-Store Experience': 'negative', 'Delivery & Orders': 'negative', 'Customer Service': 'negative'}								
25	Walmart criminals canc	Customer Service,	0.74, 0.61, 0.31	{'Customer Service': 'negative', 'Delivery & Orders': 'negative', 'In-Store Experience': 'negative'}								
26	I am writing to express	In-Store Experienc	0.83, 0.51, 0.28	{'In-Store Experience': 'negative', 'Customer Service': 'negative', 'Pricing & Billing': 'negative'}								

Fig 4.3 Confidence Scores for Aspects

Chapter 5

Results and Analysis

5.1 Performance Metrics

In sentiment analysis, model performance has to be measured in order to determine how accurately the model is predicting sentiment labels, and is typically performed through a few key metrics: Accuracy, Precision, Recall, and F1-Score.

Accuracy: Measures the proportion of correct predictions out of all predictions. It's a basic performance metric, but can be misleading for imbalanced datasets.

Precision: Evaluates the proportion of correct positive predictions among all predicted positives. It's important when false positives are costly or need to be minimized.

Recall: Focuses on how many actual positive examples the model correctly identifies. It's critical when missing positive examples (false negatives) is more problematic than misclassifying some negatives.

F1-Score: Provides a balanced measure by combining precision and recall through their harmonic mean. It's useful when both false positives and false negatives need to be minimized equally.

Confusion Matrix: A table that shows the true positives, false positives, true negatives, and false negatives, helping you understand where the model is making errors.

5.2 Comparative Analysis of Models

5.2.1 Logistic Regression

Logistic Regression

```
[28] model = LogisticRegression(max_iter=1000)
```

```
model.fit(X_train, Y_train)
```

```
LogisticRegression
LogisticRegression(max_iter=1000)
```

Evaluation for the Logistic Regression Model (Accuracy Score & the Other Performance Metrics)

```
[30] # Accuracy score on the training data

     X_train_prediction = model.predict(X_train)
     training_data_accuracy = accuracy_score( Y_train, X_train_prediction)
```

```
[31] print('Accuracy Score on the training data:', training_data_accuracy)

Accuracy Score on the training data: 0.7580615610113595
```

```
[32] # Accuracy score on the test data

     X_test_prediction = model.predict(X_test)
     test_data_accuracy = accuracy_score( Y_test, X_test_prediction)
```

```
[33] print('Accuracy Score on the test data:', test_data_accuracy)

Accuracy Score on the test data: 0.6269695859289117
```

Fig 5.1 Logistic Regression

5.2.2 CNN

```
loss, accuracy = model.evaluate(X_test, Y_test)
print(f'Test Accuracy: {accuracy * 100:.2f}%')

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```

```
86/86 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.6926 - loss: 2.0409
Test Accuracy: 68.85%
```

Fig 5.2 CNN Model Accuracy

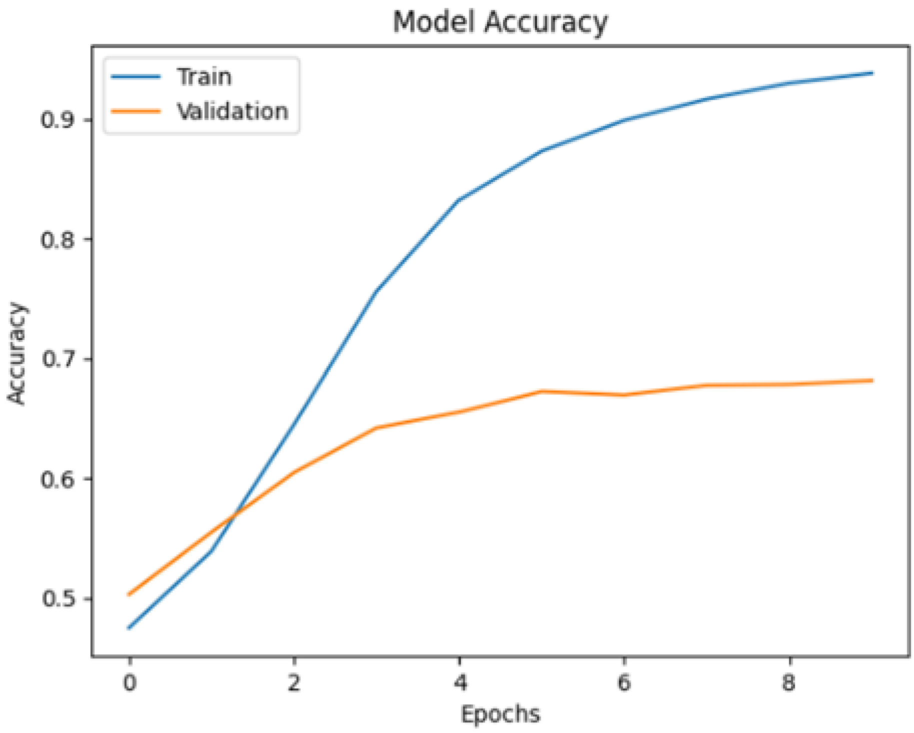


Fig 5.3 CNN Model Accuracy Vs Epochs Graph

5.2.3 LSTM

```
[ ] history = model.fit(X_train, Y_train, epochs=20, validation_data=(X_test, Y_test), callbacks=[early_stopping])

Epoch 1/20
72/72 ━━━━━━━━━━━━━━━━━━━━ 14s 119ms/step - accuracy: 0.7608 - loss: 1.1229 - val_accuracy: 0.7951 - val_loss: 0.7386
Epoch 2/20
72/72 ━━━━━━━━━━━━━━━━━━━━ 10s 117ms/step - accuracy: 0.8061 - loss: 0.7087 - val_accuracy: 0.7951 - val_loss: 0.7304
Epoch 3/20
72/72 ━━━━━━━━━━━━━━━━━━━━ 9s 95ms/step - accuracy: 0.8023 - loss: 0.7075 - val_accuracy: 0.7951 - val_loss: 0.6991
Epoch 4/20
72/72 ━━━━━━━━━━━━━━━━━━━━ 10s 90ms/step - accuracy: 0.7986 - loss: 0.7039 - val_accuracy: 0.7951 - val_loss: 0.6843
Epoch 5/20
72/72 ━━━━━━━━━━━━━━━━━━━━ 11s 95ms/step - accuracy: 0.8126 - loss: 0.6485 - val_accuracy: 0.7951 - val_loss: 0.6852
Epoch 6/20
72/72 ━━━━━━━━━━━━━━━━━━━━ 11s 108ms/step - accuracy: 0.8134 - loss: 0.6412 - val_accuracy: 0.7951 - val_loss: 0.6884
Epoch 7/20
72/72 ━━━━━━━━━━━━━━━━━━━━ 10s 109ms/step - accuracy: 0.8004 - loss: 0.6747 - val_accuracy: 0.7951 - val_loss: 0.6978
```

```
[ ] model.summary()

Model: "sequential"
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(32, 100, 16)	255,600
lstm (LSTM)	(32, 100, 64)	20,736
dropout (Dropout)	(32, 100, 64)	0
lstm_1 (LSTM)	(32, 32)	12,416
dense (Dense)	(32, 16)	528
dense_1 (Dense)	(32, 5)	85

```
Total params: 868,097 (3.31 MB)
Trainable params: 289,365 (1.10 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 578,732 (2.21 MB)
```

Fig 5.4 LSTM Model Summary

5.2.4 RoBERTa

```
train_model(model, train_loader, optimizer, criterion, device, epochs=10)
[16]
...
Epoch 1: Loss=1.1656, Accuracy=0.5726
Epoch 2: Loss=0.8363, Accuracy=0.6996
Epoch 3: Loss=0.5898, Accuracy=0.7960
Epoch 7: Loss=0.1113, Accuracy=0.9678
Epoch 8: Loss=0.0781, Accuracy=0.9769
Epoch 9: Loss=0.0512, Accuracy=0.9859
Epoch 10: Loss=0.0364, Accuracy=0.9907
```

Fig 5.5 RoBERTa Model Epochs

```
# Set model to evaluation mode
model.eval()
all_preds, all_labels = [], []

with torch.no_grad():
    for batch in test_loader:
        input_ids, attention_mask, labels = batch['input_ids'].to(device), batch['attention_mask'].to(device), batch['label'].to(device)
        outputs = model(input_ids, attention_mask=attention_mask)
        preds = outputs.logits.argmax(dim=1)

        all_preds.extend(preds.cpu().numpy())
        all_labels.extend(labels.cpu().numpy())

# Print classification report
print("Classification Report:\n", classification_report(all_labels, all_preds, target_names=label_encoder.classes_))
```

```
[17]
Classification Report:
              precision    recall  f1-score   support

       angry       0.65      0.57      0.60        23
disappointed       0.70      0.72      0.71       491
     excited       0.33      0.25      0.29         8
  frustrated       0.83      0.82      0.82      1286
       happy       0.79      0.86      0.82       144
     hopeful       0.75      0.53      0.62        17
      joyous       0.62      0.70      0.65        23
    relieved       0.77      0.77      0.77       147
         sad       0.69      0.72      0.70       348
   surprised       0.72      0.64      0.68       242

    accuracy                           0.77      2729
   macro avg       0.68      0.66      0.67      2729
weighted avg       0.77      0.77      0.77      2729
```

Fig 5.6 RoBERTa Model Classification Report

5.2.5 XLNet

```
# Load XLNet Model
xlnet_model = XLNetForSequenceClassification.from_pretrained('xlnet-base-cased', num_labels=len(label_encoder.classes_))
xlnet_model.to(device)

# Define Optimizer and Loss Function
optimizer_xlnet = optim.AdamW(xlnet_model.parameters(), lr=2e-5)

# Train XLNet
train_model(xlnet_model, train_loader_xlnet, optimizer_xlnet, criterion, device, epochs=10)

# Evaluate XLNet
xlnet_accuracy, xlnet_report = evaluate_model(xlnet_model, test_loader_xlnet, device)
print("XLNet Performance Metrics:")
print(f"Accuracy: {xlnet_accuracy}")
print(xlnet_report)
```

```
[13]
pytorch_model.bin:   0%|          | 0.00/467M [00:00<?, ?B/s]

Some weights of XLNetForSequenceClassification were not initialized from the model checkpoint at xlnet-base-cased and are ne
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
Epoch 1/10, Loss: 1.195082097852623
Epoch 2/10, Loss: 0.8680507792876317
Epoch 3/10, Loss: 0.5954666034014888
Epoch 4/10, Loss: 0.38366520413003125
Epoch 5/10, Loss: 0.2641446336233927
Epoch 9/10, Loss: 0.0984135348184355
Epoch 10/10, Loss: 0.0917182969627049
XLNet Performance Metrics:
Accuracy: 0.7537559545621106
              precision    recall  f1-score   support

       angry       0.48      0.61      0.54        23
disappointed       0.67      0.68      0.67       491
     excited       0.60      0.38      0.46         8
  frustrated       0.81      0.82      0.82      1286
       happy       0.92      0.75      0.82       144
     hopeful       0.64      0.41      0.50        17
      joyous       0.94      0.65      0.77        23
    relieved       0.69      0.86      0.77       147
         sad       0.70      0.67      0.68       348
   surprised       0.71      0.66      0.68       242

    accuracy                           0.75      2729
   macro avg       0.71      0.65      0.67      2729
weighted avg       0.76      0.75      0.75      2729
```

Fig 5.7 XLNet Model Epochs and Classification Report

5.2.6 Hybrid Model (BERTBiLSTMAttention)

```
Epoch 1/20 - Train Acc: 0.1036 | Val Acc: 0.1198
Model improved and saved.
Epoch 2/20 - Train Acc: 0.2556 | Val Acc: 0.5255
Model improved and saved.
Epoch 3/20 - Train Acc: 0.3874 | Val Acc: 0.4771
No improvement. Patience 1/5
Epoch 4/20 - Train Acc: 0.4388 | Val Acc: 0.5665
Model improved and saved.
Epoch 5/20 - Train Acc: 0.4897 | Val Acc: 0.4756
No improvement. Patience 1/5
Epoch 6/20 - Train Acc: 0.5627 | Val Acc: 0.6043
Model improved and saved.
Epoch 7/20 - Train Acc: 0.6632 | Val Acc: 0.5977
No improvement. Patience 1/5
Epoch 8/20 - Train Acc: 0.7362 | Val Acc: 0.5486
No improvement. Patience 2/5
Epoch 9/20 - Train Acc: 0.7990 | Val Acc: 0.6838
Model improved and saved.
Epoch 10/20 - Train Acc: 0.8385 | Val Acc: 0.7479
Model improved and saved.
Epoch 11/20 - Train Acc: 0.8791 | Val Acc: 0.7501
Model improved and saved.
Epoch 12/20 - Train Acc: 0.9064 | Val Acc: 0.7706
Model improved and saved.
Epoch 13/20 - Train Acc: 0.9294 | Val Acc: 0.7761
...
    accuracy                           0.80      2729
   macro avg       0.78      0.68      0.71      2729
weighted avg       0.79      0.80      0.79      2729
```

Fig 5.8 Hybrid Model Epochs

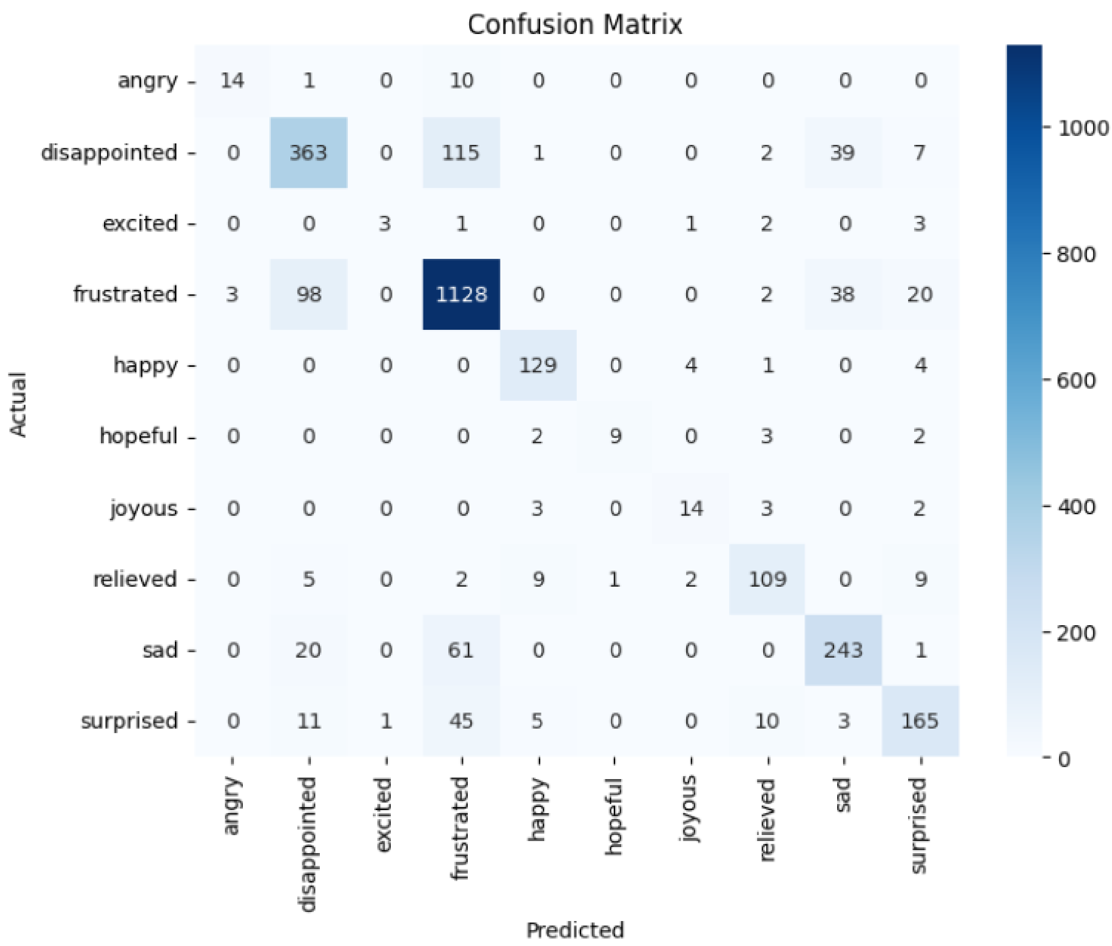
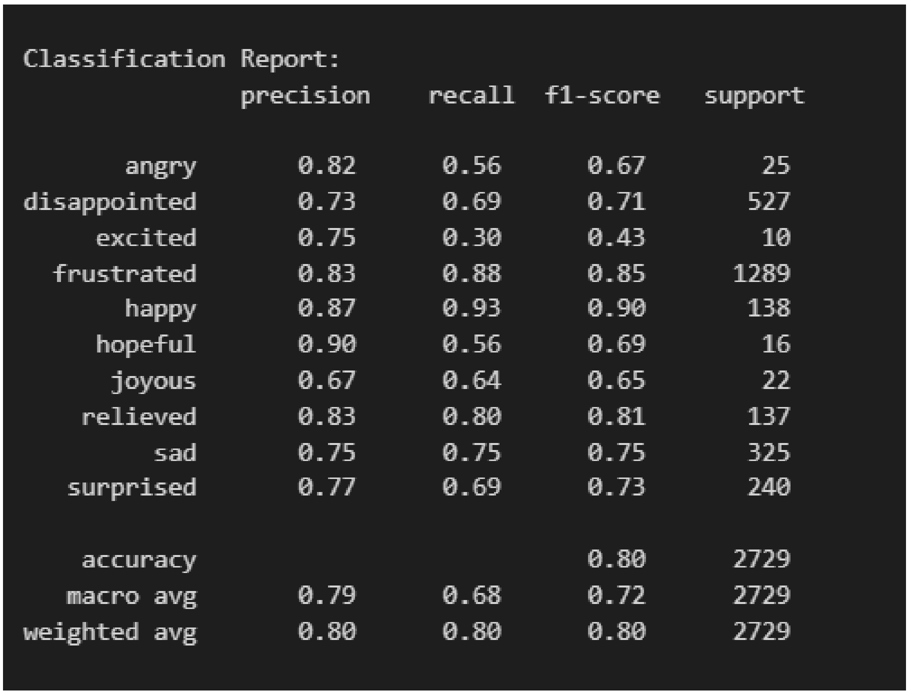


Fig 5.9 Hybrid Model Confusion Matrix



Classification Report:

	precision	recall	f1-score	support
angry	0.82	0.56	0.67	25
disappointed	0.73	0.69	0.71	527
excited	0.75	0.30	0.43	10
frustrated	0.83	0.88	0.85	1289
happy	0.87	0.93	0.90	138
hopeful	0.90	0.56	0.69	16
joyous	0.67	0.64	0.65	22
relieved	0.83	0.80	0.81	137
sad	0.75	0.75	0.75	325
surprised	0.77	0.69	0.73	240
accuracy			0.80	2729
macro avg	0.79	0.68	0.72	2729
weighted avg	0.80	0.80	0.80	2729

Fig 5.10 Hybrid Model Classification Report

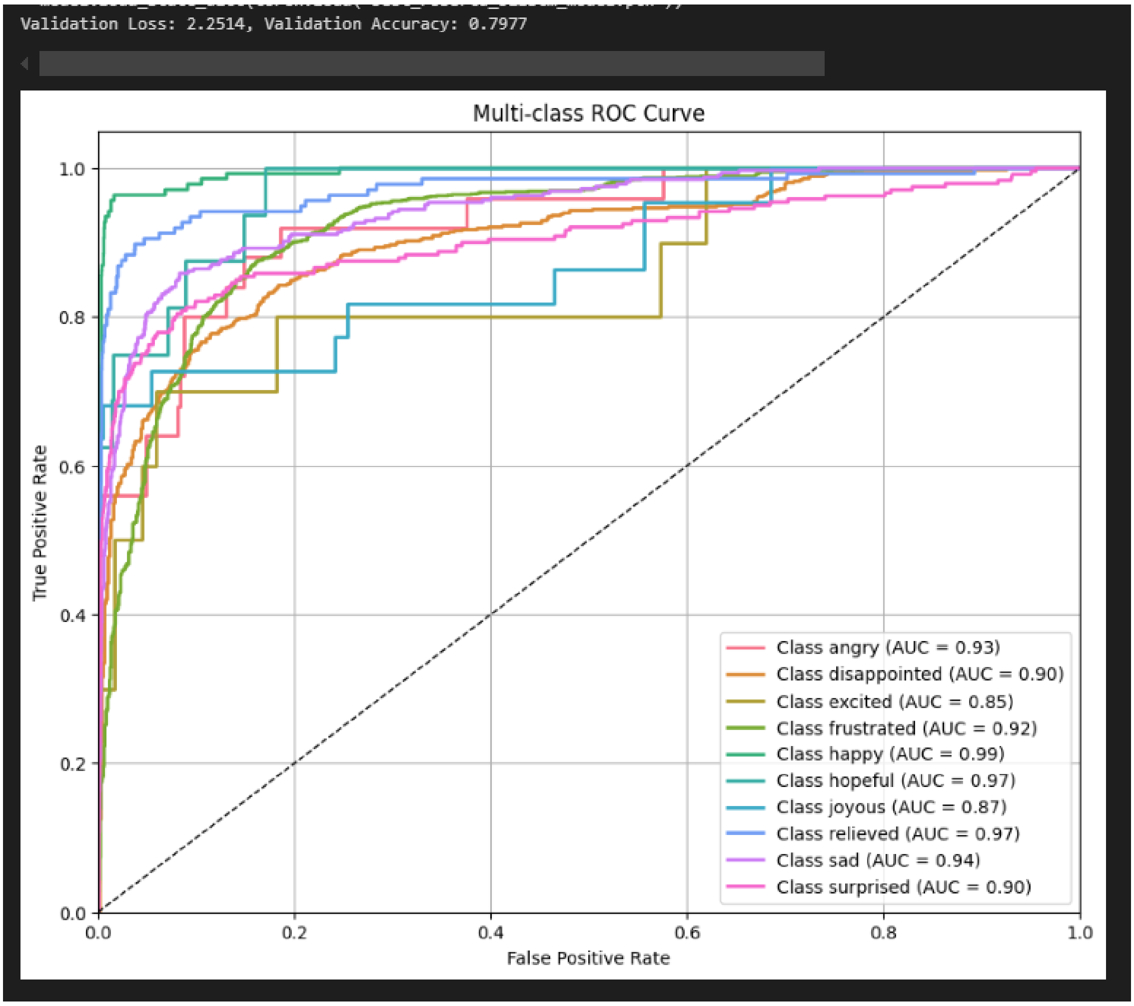


Fig 5.11 Hybrid Model Multi-Class ROC Curve

```
# Convert to numpy arrays
all_preds = np.array(all_preds)
all_labels = np.array(all_labels)

# Per-Class Accuracy
print("\nPer-Class Accuracy:")
for class_idx in range(num_classes):
    class_name = id2label[class_idx]
    idx = (all_labels == class_idx)
    acc = accuracy_score(all_labels[idx], all_preds[idx])
    print(f"{class_name}: {acc:.4f}")
```

```
/tmp/ipykernel_31/1110184285.py:4: FutureWarning: You are using
  model.load_state_dict(torch.load("best_roberta_bilstm_model.pt

Per-Class Accuracy:
angry: 0.5600
disappointed: 0.6888
excited: 0.3000
frustrated: 0.8751
happy: 0.9348
hopeful: 0.5625
joyous: 0.6364
relieved: 0.7956
sad: 0.7477
surprised: 0.6875
```

Fig 5.12 Hybrid Model Per-Class Accuracy

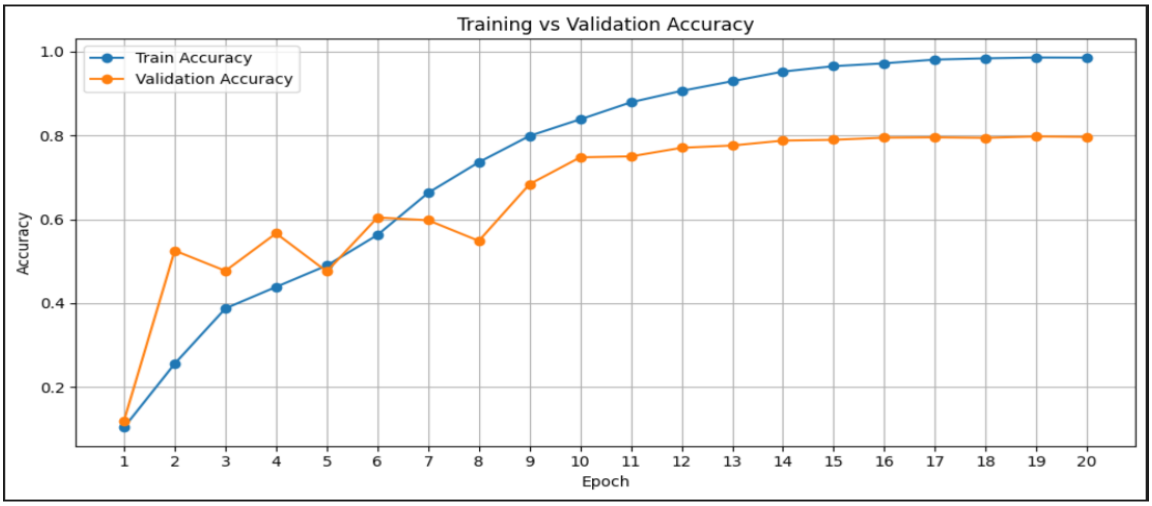


Fig 5.13 Hybrid Model Training Vs Validation Accuracy Graph

COMPARISON:

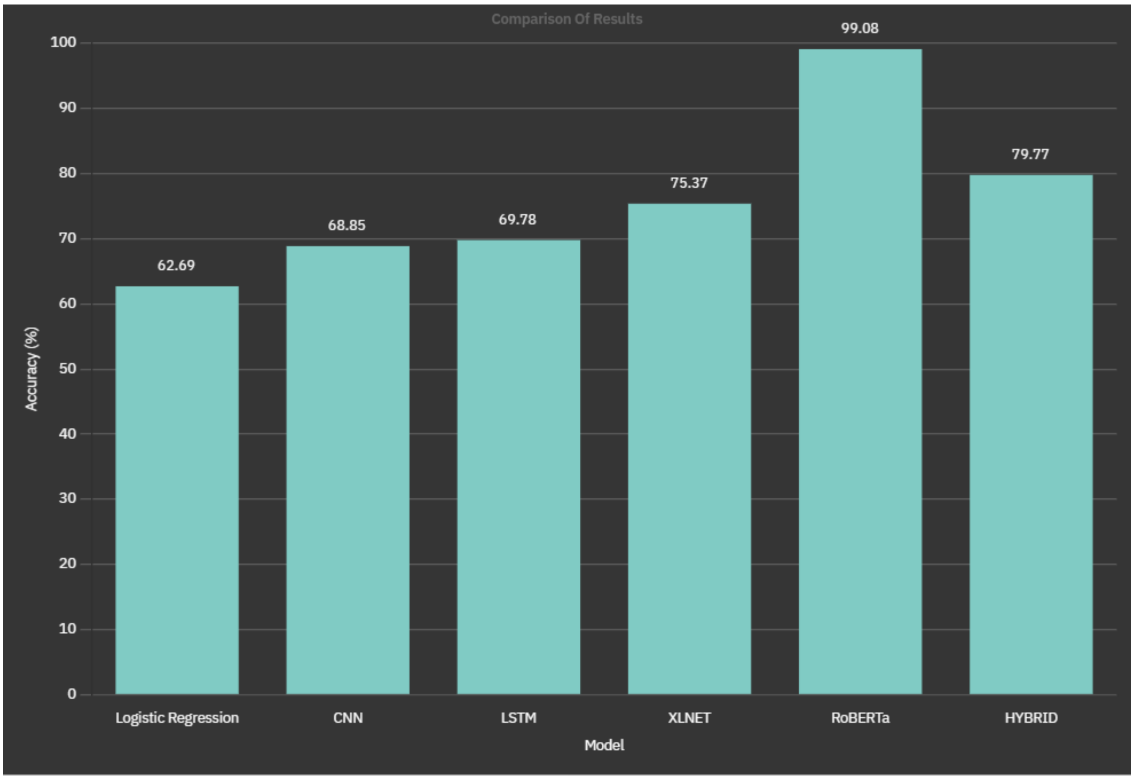


Fig 5.14 Comparison of Models

5.3 Sentiment Distribution Across Aspects

When carrying out aspect-based sentiment analysis (ABSA), one must maintain sentiment distribution across aspects or discussion topics in customer comments. Aspects may be different dimensions of a service, e.g., Customer Service, Delivery & Orders, Pricing & Billing, In-Store Experience, etc. By observing the sentiment division between both these groups, we can ascertain relevant information based on how customers perceive various elements of the product or service. For instance, a customer would be positively biased towards the delivery process and negatively biased towards the customer service process.

Having done the sentiment analysis, the next thing to do is to examine to what extent sentiments (negative, positive, or neutral) coincide each section. Segmentation can be expressed in various forms, for example, as bar charts or pie charts showing the extent of positive, negative, and neutral sentiment for each topic. For example, if in Customer Service always its feedback is having very high negative sentiments, then there it should be the area of improvement. But if In-Store Experience or Delivery & Orders are having positive sentiments overall, it indicates that these attributes are doing good from the customers' perspective.

Sentiment breakdown also gives some notion of probable contradictions or opposite sentiments being used in the reviews. For example, the customers may like the product but be dissatisfied with the Pricing & Billing or Order Delivery Time, citing particular areas where the company needs to get better. Sentiment distribution analysis enables the companies to keep themselves abreast of trends, identify hotspots of problems, and direct resources accordingly for improving customer experience in most effective ways.

5.4 Case Study of Walmart Services Sentiment

Customer sentiments in reviews are first classified into positive, negative, and neutral opinions through models like BERT or RoBERTa in this study. For example, customers can rant on the Customer Service category about too sluggish responses or rude staff, creating unwarranted negativity on this category. Posts on Pricing & Billing will have more scattered sentiment alignment because customers appreciate low prices from Walmart but complain about fees or unexpected charges. Delivery & Orders can emphasize customer satisfaction at ordering ease with online buying but when delays in delivery or wrong products are experienced, more adverse or neutral opinions can be found in this category.

Once sentiment classification is completed, additional analysis can be carried out to determine the percentage of positive, negative, and neutral opinions in different dimensions. These visualization techniques such as heatmaps or bar charts can be utilized in order to simply represent the sentiment trend and figure out which aspect of Walmart's service requires work. If, for instance, customer sentiment regarding In-Store Experience is overwhelmingly positive in number, then as whole customers are satisfied with in-store experience, and a surge of negative reviews on Delivery & Orders would prompt Walmart to revisit its shipping.

This case study also allows Walmart to track sentiment over time. Comparing pre- and post-change sentiment distributions across individual changes, for example, a redesign of the Customer Service department or price strategy adjustments, Walmart can assess whether its efforts to better its certain segments have been worth it. In addition, by conducting this case study, Walmart can gain actionable customer satisfaction information that it can use in designing future service improvement and customer loyalty development initiatives. Lastly, this exhaustive sentiment analysis provides Walmart with an immediate understanding of its weaknesses and strengths from the customer's perspective, enabling the company to address those areas directly impacting consumer satisfaction.

5.5 Summary of Findings

Through the comparative study of some of the ML and DL models for Walmart customer review sentiment classification indicates notable differences in performance. Of all the examined models, RoBERTa had the best accuracy at 99.08%, far outperforming all other methods. The hybrid model (Bi-LSTM + BERT + Attention) placed second at 79.77%, marginally ahead of XLNet at 75.37%. LSTM (69.78%) and CNN (68.85%) were modestly performing traditional DL models, while the simplest ML Model, Logistic Regression, achieved the lowest accuracy of 62.69%. These findings validate that transformer models and hybrid models perform much better in identifying sophisticated patterns in text sentiment data than classical ML models.

5.7 Limitations of the Study

Despite the success of this study in employing state-of-the-art NLP techniques for sentiment analysis of Walmart services, there are some limitations. First of all, the models used fail to accurately captures sarcasm, irony, or nuanced emotional statements, leading to potential misclassifications of reviews. The application of pre-trained models like Facebook/BART-Large-MNLI and CardiffNLP/Twitter-RoBERTa, which are not tailored to the retail sector or Walmart-specific vocabulary, further limits accuracy in detecting the context-specific sentiments.

The AER process is restricted to predetermined categories, thus the model will tend to ignore underlying or less common aspects customers discussed. Further, imbalanced distribution of data between sentiment classes distorts model performance towards highly frequent emotions. The web scraping technique used to obtain data also creates issues, i.e., changing website structure, potential biasness in available reviews, and ethical concerns.

Lastly, deep learning models computation is challenging to apply, making scaling the system to real-time or large-scale use difficult. These limits indicate a need for additional customization, fine-tuning, and optimization in future work.

Chapter 6

Conclusion and Future Work

6.1 Challenges Involved

The sentiment analysis of Walmart services is riddled with several challenges, primarily in text data management of large size, sentiment classification, and extracting meaningful insights from the customer reviews. Data collection and preprocessing are among the primary challenges since it involves web scraping of customer reviews through web scraping on Trustpilot, which means handling dynamic webpage layouts, rate limiting, and data cleaning to remove duplicate or unwanted records. Another major challenge is handling ambiguity and sarcasm, as sentiment analysis software tends to struggle to handle sarcastic comments where a seemingly positive phrase can actually have negative sentiment (e.g., "Great job Walmart, my package lost again!"). Second, multi-aspect sentiment analysis adds complexity because it entails locating specific areas of Walmart's service, e.g., customer service, delivery, in-store experience, and price, and labelling sentiment polarity per area. Unbalanced distribution of data is yet another area of concern because certain categories or sentiments are hardly getting any reviews, and hence provide biased results. There is also an issue of domain adaptation, as pre-trained models like Facebook/BART-Large-MNLI and CardiffNLP/Twitter-RoBERTa-base-sentiment-latest may not be fully capable of grasping retail-specific jargon and customer reviews.

Lastly, computational complexity is a challenge while employing deep learning models such as LSTM, RoBERTa, and Bi-LSTM + BERT + Attention because of the heavy processing required to train gigantic models with gigantic datasets that requires high optimisation power alongside computation for quality accuracy. Breaking such barriers is necessary so as to achieve true and actionable results out of sentiment analysis for retailing.

6.2 Future Enhancements

Since the current Walmart service sentiment analysis demonstrates strong results, there are several promising directions for future enhancement to improve its accuracy, scalability, and overall impact.

First, expanding the dataset to include reviews from multiple platforms such as Walmart's website, Reddit, Twitter, Facebook, and Google Reviews would reduce sampling bias and offer a broader sentiment perspective. Additionally, tracking time-based sentiment trends would help identify seasonal customer satisfaction patterns or spikes in dissatisfaction.

Aspect-Based Sentiment Analysis (ABSA) could be advanced by using unsupervised or semi-supervised techniques like Latent Dirichlet Allocation or BERT-based topic modeling, enabling automatic discovery of emerging subtopics instead of relying on fixed categories. For emotion and sentiment classification, fine-tuning transformer models specifically on Walmart-related customer data could improve contextual understanding, especially for nuanced expressions like sarcasm or mixed sentiments.

Exploring multimodal sentiment analysis by incorporating audio and video reviews from platforms like YouTube or TikTok or Instagram Reels would add much better insights through tone, expression, and delivery. Moreover, developing real-time sentiment analysis systems would allow Walmart to proactively detect and respond to customer concerns, even predicting issues such as delivery delays using sentiment trends.

From a ML standpoint, while the hybrid Bi-LSTM + BERT + Attention model performed well, adopting newer models like T5, GPT-4, or LongFormer could enhance the handling of long and complex reviews.

Lastly, expanding to multilingual sentiment analysis using models like XLM-RoBERTa or multilingual BERT would ensure insights from non-English-speaking customers are included, making the analysis more globally representative.

In conclusion, while the research has proven the effectiveness of sentiment analysis for evaluating Walmart's services, incorporating diverse data sources, advanced NLP, real-time and multilingual capabilities will significantly deepen its relevance and utility for future applications.

6.3 Final Thoughts

We effectively performed sentiment analysis of Walmart services using natural language processing (NLP) methods, zero-shot classification, aspect-based sentiment analysis (ABSA), and deep learning models. From Trustpilot customer review extraction and sentiment analysis, we labelled sentiments into ten emotion labels and provided aspect-based sentiment scores for four major aspects: Customer Service, Delivery & Orders, In-Store Experience, and Pricing & Billing. The application of deep learning and advanced machine learning models such as Logistic Regression, CNN, LSTM, RoBERTa, XLNet, and a Bi-LSTM+BERT+Attention hybrid model identified Walmart's services strengths and weaknesses from the customer's point of view. The results of this research provide evidence of the presence of huge customer pain points and positive moments, which allow Walmart to improve areas and increase customer satisfaction.

For instance, delivery problems emerged as a major issue, negative sentiment dominating there, while the in-store experience was largely favourable, that supported Walmart retailing. It also confirmed that transformer-based models were superior to conventional ML models and that the hybrid Bi-LSTM+BERT+Attention model was the best overall to detect sentiment. The study was subject to some limitations such as data source limitations, possibility of zero-shot misclassifications, difficulty in handling sarcasm and implicit sentiment, and computational limitations of deep learning models. Future studies must take into account expansion of data sources, improvement of aspect extraction method, incorporation of real-time sentiment monitoring, and investigation of multilingual sentiment analysis in developing a more scalable and robust sentiment analysis system. Overall, this study reaffirms the effectiveness of sentiment analysis in determining customer experiences and places emphasis on the process whereby knowledge obtained using data can drive organizations to take informed decisions.

As the companies like Walmart will continue enhancing their services, the application of AI-based sentiment analysis will remain an effective methodology for determining the sentiments of the customers, refining processes, and creating customer loyalty.

By continuously fine-tuning such processes and broadening the analysis scope, companies can look forward to anticipating customer problems, creating service excellence, and sparking lasting customer satisfaction.

References

[1] Jahanzeb Jabbar, Iqra Urooj, Wu JunSheng, and Naqash Azeem, "Real-time Sentiment Analysis On E-Commerce Application," Proceedings of the 2019 IEEE 16th International Conference on Networking, Sensing and Control, May 9-11, 2019, Banff, Alberta, Canada. Available at [IEEE Xplore].

[2] Shrey Arora, Siddharth Mahapatra, Dr. Anil Jadav, Manojeet Barla, and Nishant Mallick, "Temporal and Sentimental Analysis of Customer Reviews," Proceedings of the 2024 IEEE 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence), January 2024, Pune, India. Available at [IEEE Xplore].

[3] M. S. A and P. J. B. Princess, "Aspect-Based Sentiment Analysis with Triplet Extraction for E-Commerce," 2024 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 2024, pp. 1082-1087, doi: 10.1109/ICICT60155.2024.10544958.

[4] E. Lee, E. Wahyudi, and R. Kusumaningrum, "Aspect-Based Sentiment Analysis in E-Commerce User Reviews Using Latent Dirichlet Allocation (LDA) and Sentiment Lexicon," 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 2019, pp. 1-6, doi: 10.1109/ICICoS48119.2019.8982522.

[5] S. Zulfiker, A. Chowdhury, D. Roy, S. Datta, and S. Momen, "Bangla E-Commerce Sentiment Analysis Using a Machine Learning Approach," 2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh, 2022, pp. 1-5, doi: 10.1109/STI56238.2022.10103350.

[6] G. P. M. S., P. G. J., R. Kumar, P. K. H. R., V. S. K., and A. Dahiya, "Sentiment Analysis Using Transfer Learning for E-Commerce Websites," 2024 International Conference on Advances in Modern Age Technologies for Health and Engineering Science (AMATHE), Shivamogga, India, 2024, pp. 1-5, doi: 10.1109/AMATHE61652.2024.10582143.

[7] P. K. Bharti, P. Kumar, M. K. Sharma, A. A. Hassan, and M. Zaid, "Review of Efficient Methods for Sentiment Analysis of E-Commerce Customer Review using Machine Learning Techniques," 2023 5th International Conference on Advances in

Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2023, pp. 204-208, doi: 10.1109/ICAC3N60023.2023.10541681.

[8] D. Chaudhari, S. Angadi, S. Sable, U. Patil, D. Chaudhari, and K. Jadhav, "Sentimental Analysis on Zomato Restaurant Reviews using Bi-LSTM," 2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS), Kalaburagi, India, 2023, pp. 1-7, doi: 10.1109/ICIICS59993.2023.10421062.

[9] M. Hakkinen, F. A. Wong, M. S. Anggreainy, and W. R. Hidayat, "Sentiment Analysis of E-commerce Review using Lexicon Sentiment Method," 2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI), Jakarta, Indonesia, 2021, pp. 68-71, doi: 10.1109/ICCSAI53272.2021.9609786.

[10] P. Kathuria, P. Sethi, and R. Negi, "Sentiment analysis on E-commerce reviews and ratings using ML & NLP models to understand consumer behavior," 2022 International Conference on Recent Trends in Microelectronics, Automation, Computing and Communications Systems (ICMACC), Hyderabad, India, 2022, pp. 1-5, doi: 10.1109/ICMACC54824.2022.10093674.

[11] M. Al Omari, M. Al-Hajj, N. Hammami, and A. Sabra, "Sentiment Classifier: Logistic Regression for Arabic Services' Reviews in Lebanon," 2019 International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 2019, pp. 1-5, doi: 10.1109/ICCISci.2019.8716394.

[12] A. K. Bitto, M. S. Arman, R. M. Saha, H. Jahan, I. Mahmud and A. Das, "Customer Sentiments Towards Delivery Services in Bangladesh: A Machine Learning-Based Sentiment Analysis," *2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS)*, Dhaka, Bangladesh, 2024, pp. 1-5, doi: 10.1109/iCACCESS61735.2024.10499618.

[13] Puspita Kencana Sari et al 2018 J. Phys.: Conf. Ser. 971 012053, DOI 10.1088/1742-6596/971/1/012053. 2.1.14 P. Mugil, C. Jeevanantham, Y. N. N. Yadav, M. Harikaran. and R. Dhanapal, "A Secure and Efficient Product Analysis for Trust based E-Commerce Application," 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2023, pp. 1370-1376, doi: 10.1109/ICSCDS56580.2023.10104925.

[14] P. Mugil, C. Jeevanantham, Y. N. N. Yadav, M. Harikaran. and R. Dhanapal, "A Secure and Efficient Product Analysis for Trust based E-Commerce Application," 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2023, 1370-1376, doi: 10.1109/ICSCDS56580.2023.10104925.

[15] J. Panduro-Ramirez, "Sentiment Analysis in Customer Reviews for Product Recommendation in E-commerce Using Machine Learning," 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2024, pp. 1-5, doi: 10.1109/ACCAI61061.2024.10602027.

[16] O. Taneja, K. R. Satya Alpana, D. C. Rathod, A. F. M R and V. P. H, "Unraveling Customer Sentiment in Indian Fashion E-Commerce: A BERT-Based Approach," 2024 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 2024, pp. 1-5, doi: 10.1109/CSITSS64042.2024.10816956.

[17] M. M. B. Mboungou, I. Yamin, S. Zhang and A. Iqbal, "Sentiment Analysis of Client Reviews on a French E-Commerce Platform," 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), Ballari, India, 2023, pp. 1-6, doi: 10.1109/AIKIIE60097.2023.10390055.

[18] M. A. Casas-Valadez, A. Faz-Mendoza, C. E. Medina-Rodríguez, R. Castañeda-Miranda, N. K. Gamboa-Rosales and J. R. López-Robles, "The synergic relationship between e-commerce and Sentiment Analysis: A content analysis of published articles between 2007 and 2020," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, 2020, pp. 1-5, doi: 10.1109/ICDABI51230.2020.9325689.

[19] D. Jahnavi, S. Sami, S. Pulata, G. Bharathi Mohan and R. Prasanna Kumar, "Robust Hybrid Model for Social Media Sentiment Analysis," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-6, doi: 10.1109/ICCCNT61001.2024.10725264.

[20] M. J. C. Samonte, A. T. G. Dela Rosa, L. J. C. Rivera and J. S. E. Silo, "Using Hybrid CNN-LSTM Model for Sentiment Analysis of COVID-19 Tweets," 2023 13th International Conference on Software Technology and Engineering (ICSTE), Osaka, Japan, 2023, pp. 133-142, doi: 10.1109/ICSTE61649.2023.00029.

Appendix: (Codebase)

```
import cv2

import dlib
import numpy as np
import screen_brightness_control as sbc
from plyer import notification
import time
import platform
import subprocess
from scipy.spatial import distance as dist
import logging
import sounddevice as sd
import soundfile as sf
import time

from time import sleep
import requests
import pandas as pd
from bs4 import BeautifulSoup

# Function to extract data from BeautifulSoup objects
def soup2list(src, list_, attr=None):
    if attr:
        for val in src:
            list_.append(val.get(attr, "N/A"))  # Avoid KeyError if attribute is missing
    else:
        for val in src:
            list_.append(val.get_text(strip=True))  # Strip spaces

# Lists to store extracted data
users = []
userReviewNum = []
ratings = []
locations = []
dates = []
reviews = []

from_page = 111
to_page = 411
company = 'www.walmart.com'

headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 
(KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36"
}  # Added headers to prevent request blocking
```

```
for i in range(from_page, to_page + 1):
    url = f"https://www.trustpilot.com/review/{company}?page={i}"
    result = requests.get(url, headers=headers)

    if result.status_code != 200:
        print(f"Failed to retrieve page {i}. Status code: {result.status_code}")
        continue  # Skip this page if request fails

    soup = BeautifulSoup(result.content, "html.parser")

    # Extracting Usernames
    soup2list(soup.find_all('span', class_='typography_heading-xs__osRhC'), users)  #
Usernames

    # Extracting Total Reviews (Numbers only)
    for user_info in soup.find_all('div', class_='styles_consumerExtraDetails__TylYM'):
        total_reviews_text = user_info.get_text(strip=True)
        total_reviews = ''.join(filter(str.isdigit, total_reviews_text))  # Extract digits only
        userReviewNum.append(total_reviews if total_reviews else "N/A")

    # Extracting Locations (Only country, excluding review count)
    soup2list(soup.find_all('span', class_='typography_body-m__k2UI7
typography_appearance-subtle__PYOVM'), locations)

    # Extracting Dates
    soup2list(soup.find_all('time'), dates, attr='datetime')  # Dates

    # Extracting Ratings
    soup2list(soup.find_all('div', class_='styles_reviewHeader__PuHBd'), ratings,
attr='data-service-review-rating')  # Ratings

    # Extracting Review Texts
    for review_content in soup.find_all('div', class_='styles_reviewContent__SCYfD'):
        review_text = review_content.get_text(strip=True)
        reviews.append(review_text if review_text else "N/A")

    sleep(1)  # Prevents request throttling

# Fix: Ensure all lists have the same length
max_length = max(len(users), len(userReviewNum), len(locations), len(dates),
len(reviews), len(ratings))

# Pad shorter lists with "N/A"
users += ["N/A"] * (max_length - len(users))
userReviewNum += ["N/A"] * (max_length - len(userReviewNum))
locations += ["N/A"] * (max_length - len(locations))
dates += ["N/A"] * (max_length - len(dates))
reviews += ["N/A"] * (max_length - len(reviews))
```

```
ratings += ["N/A"] * (max_length - len(ratings))

# Creating a DataFrame
review_data = pd.DataFrame(
    {
        'Username': users,
        'Total reviews': userReviewNum,
        'Location': locations,
        'Date': dates,
        'Review': reviews,
        'Rating': ratings
    }
)

# # Optionally save to CSV
# review_data.to_csv("review_data.csv", index=False)

# Display first 5 rows for debugging purposes
print(review_data.head())
```

```
import pandas as pd

file_path = "/kaggle/input/reviews-
dataset/zero_shot_labeled_reviews_final.csv"
df = pd.read_csv(file_path)
from transformers import pipeline

aspect_extractor = pipeline("zero-shot-classification",
model="facebook/bart-large-mnli")
# Walmart service aspects based on my reviews dataset

aspect_labels = [
    "Customer Service",   # Issues related to support, responses, and
resolutions
    "Delivery & Orders",  # Delivery speed, order issues, missing items
    "In-Store Experience",  # Cleanliness, waiting time, stock
availability
    "Pricing & Billing",  # Pricing complaints, refunds, extra charges
]
# Extract aspects using zero-shot classification

aspect_results = {review: aspect_extractor(review, aspect_labels,
multi_label=True) for review in sample_reviews}
# Show a few results

for review, result in list(aspect_results.items())[:5]:
    print(f"Review: {review}")
```

```
    print(f"Predicted Aspects: {result['labels'][:3]}")  # Shows the top
3 predicted aspects
    print(f"Confidence Scores: {result['scores'][:3]}")
    print("-" * 80)
# Load Zero-Shot Classification Model (BART-MNLI)

aspect_extractor = pipeline("zero-shot-classification",
model="facebook/bart-large-mnli", device=0)  # Use GPU (cuda:0)

# Run Aspect Extraction on all the Reviews

reviews = df["Review"].tolist()
aspect_results = [aspect_extractor(review, aspect_labels,
multi_label=True) for review in reviews]
# Convert the Results to DataFrame

aspect_data = []
for review, result in zip(reviews, aspect_results):
    aspect_data.append({
        "review": review,
        "predicted_aspects": ", ".join(result["labels"][:3]),  # Top 3
predicted aspects
        "confidence_scores": ", ".join([str(round(score, 2)) for score in
result["scores"][:3]])
    })
# Save the Processed Data

aspect_df.to_csv("/kaggle/working/aspect_extracted_reviews.csv",
index=False)


# Load Improved RoBERTa Sentiment Model

sentiment_analyzer = pipeline("sentiment-analysis",
model="cardiffnlp/twitter-roberta-base-sentiment-latest", device=0)  #
Use latest model

tokenizer = AutoTokenizer.from_pretrained("cardiffnlp/twitter-roberta-
base-sentiment-latest")
# Aspect-Based Sentiment Analysis

def analyze_aspect_sentiments(row, debug=False, max_debug_samples=5):
    if pd.isna(row["predicted_aspects"]):  # Handle missing aspect data
        return {}

    aspects = row["predicted_aspects"].split(", ")  # Convert aspect
string to list
    aspect_sentiments = {}
```

```
    for aspect in aspects:
        review_text = f"{aspect}: {row['review']}"  # Aspect-focused text
        review_chunks = split_review(review_text, max_length=400)

        chunk_sentiments = []
        for i, chunk in enumerate(review_chunks):
            # Print debug info
            if debug and getattr(row, "name", -1) < max_debug_samples:
                print(f"[DEBUG] Review {getattr(row, 'name', -1)}: Chunk
{i+1} length = {len(tokenizer.encode(chunk))}")

            # Run Sentiment Analysis
            sentiment_result = sentiment_analyzer(chunk,
truncation=True)[0]
            chunk_sentiments.append(sentiment_result["label"])

        # Majority Vote or Default Sentiment
        final_sentiment = max(set(chunk_sentiments),
key=chunk_sentiments.count) if chunk_sentiments else "neutral"
        aspect_sentiments[aspect] = final_sentiment

    return aspect_sentiments
# Apply sentiment analysis to all reviews

df["aspect_sentiments"] = df.apply(analyze_aspect_sentiments, axis=1)


import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.stem.porter import PorterStemmer
nltk.download('stopwords')
from nltk.corpus import stopwords
STOPWORDS = set(stopwords.words('english'))

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import accuracy_score
```

```
from sklearn.linear_model import LogisticRegression
from wordcloud import WordCloud
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
import pickle
import re
#Load the data

data = pd.read_csv("/content/reviews1.csv", encoding ='ISO-8859-1')
column_names = ['username', 'totalreviews', 'location', 'date', 'review',
'rating', 'feeling']
data = pd.read_csv("/content/reviews1.csv", names = column_names,
encoding ='ISO-8859-1')

X = data['stemmed_content'].values
Y = data['feeling'].values
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =
0.2, random_state = 2)
vectorizer = TfidfVectorizer()

X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)
model = LogisticRegression(max_iter=1000)
model.fit(X_train, Y_train)
# Accuracy score on the training data

X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score( Y_train, X_train_prediction)
# Accuracy score on the test data

X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score( Y_test, X_test_prediction)
print('Accuracy Score on the test data:', test_data_accuracy)


model = Sequential([
    Embedding(input_dim=MAX_NUM_WORDS, output_dim=EMBEDDING_DIM,
input_length=MAX_SEQUENCE_LENGTH),
    Conv1D(filters=128, kernel_size=5, activation='relu'),
    GlobalMaxPooling1D(),
    Dense(64, activation='relu'),
    Dropout(0.5),
    Dense(4, activation='softmax')  # 4 output classes
])
# Compile the model

model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

```
# Train the model

model.fit(X_train, Y_train, epochs=10, batch_size=32,
validation_data=(X_test, Y_test))
# Evaluate the model

test_loss, test_acc = model.evaluate(X_test, y_test)
print(f"Test Accuracy: {test_acc:.4f}")

#XLNET

import pandas as pd
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import Dataset, DataLoader
from transformers import BertTokenizer, BertForSequenceClassification,
XLNetTokenizer, XLNetForSequenceClassification
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report

# Load the dataset
df = pd.read_csv("/kaggle/input/review-data/reviews_final.csv")

# Select relevant columns
df = df[['Cleaned_Review', 'Emotion']].dropna()

# Split dataset into train and test sets
train_texts, test_texts, train_labels, test_labels = train_test_split(
    df['Cleaned_Review'].tolist(), df['Emotion'].tolist(), test_size=0.2,
random_state=42)

# Encode emotion labels
label_encoder = LabelEncoder()
train_labels_encoded = label_encoder.fit_transform(train_labels)
test_labels_encoded = label_encoder.transform(test_labels)
class ReviewDataset(Dataset):
    def __init__(self, texts, labels, tokenizer, max_length=128):
        self.texts = texts
        self.labels = labels
        self.tokenizer = tokenizer
        self.max_length = max_length

    def __len__(self):
        return len(self.texts)

    def __getitem__(self, idx):
```

```
        encoding = self.tokenizer(
            self.texts[idx],
            truncation=True,
            padding='max_length',
            max_length=self.max_length,
            return_tensors='pt'
        )
        return {
            'input_ids': encoding['input_ids'].squeeze(0),
            'attention_mask': encoding['attention_mask'].squeeze(0),
            'labels': torch.tensor(self.labels[idx], dtype=torch.long)
        }

batch_size = 8

# Load tokenizers
bert_tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
xlnet_tokenizer = XLNetTokenizer.from_pretrained('xlnet-base-cased')

# Create datasets
train_dataset_bert = ReviewDataset(train_texts, train_labels_encoded,
bert_tokenizer)
test_dataset_bert = ReviewDataset(test_texts, test_labels_encoded,
bert_tokenizer)
train_dataset_xlnet = ReviewDataset(train_texts, train_labels_encoded,
xlnet_tokenizer)
test_dataset_xlnet = ReviewDataset(test_texts, test_labels_encoded,
xlnet_tokenizer)

# Create DataLoaders
train_loader_bert = DataLoader(train_dataset_bert, batch_size=batch_size,
shuffle=True)
test_loader_bert = DataLoader(test_dataset_bert, batch_size=batch_size,
shuffle=False)
train_loader_xlnet = DataLoader(train_dataset_xlnet,
batch_size=batch_size, shuffle=True)
test_loader_xlnet = DataLoader(test_dataset_xlnet, batch_size=batch_size,
shuffle=False)

def train_model(model, train_loader, optimizer, criterion, device,
epochs=10):
    model.train()
    for epoch in range(epochs):
        total_loss = 0
        for batch in train_loader:
            optimizer.zero_grad()
```

```
            input_ids, attention_mask, labels =
batch['input_ids'].to(device), batch['attention_mask'].to(device),
batch['labels'].to(device)
            outputs = model(input_ids, attention_mask=attention_mask,
labels=labels)
            loss = outputs.loss
            loss.backward()
            optimizer.step()
            total_loss += loss.item()
        print(f"Epoch {epoch+1}/{epochs}, Loss:
{total_loss/len(train_loader)}")

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# Load BERT Model
bert_model = BertForSequenceClassification.from_pretrained('bert-base-
uncased', num_labels=len(label_encoder.classes_))
bert_model.to(device)

# Define Optimizer and Loss Function
optimizer = optim.AdamW(bert_model.parameters(), lr=0.0001)
criterion = nn.CrossEntropyLoss()

# Train BERT
train_model(bert_model, train_loader_bert, optimizer, criterion, device,
epochs=10)

# Evaluate BERT
bert_accuracy, bert_report = evaluate_model(bert_model, test_loader_bert,
device)
print("BERT Performance Metrics:")
print(f"Accuracy: {bert_accuracy}")
print(bert_report)


#ROBERTA
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader, Dataset
from transformers import RobertaTokenizer,
RobertaForSequenceClassification, get_scheduler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report
import pandas as pd
import nlpaug.augmenter.word as naw
# Encode emotion labels
```

```
label_encoder = LabelEncoder()
df['Emotion'] = label_encoder.fit_transform(df['Emotion'])

# Load RoBERTa tokenizer
tokenizer = RobertaTokenizer.from_pretrained('roberta-base')
class EmotionDataset(Dataset):
    def __init__(self, texts, labels):
        self.texts = texts
        self.labels = labels

    def __len__(self):
        return len(self.texts)

    def __getitem__(self, idx):
        text = self.texts[idx]
        inputs = tokenizer(text, padding="max_length", truncation=True,
max_length=128, return_tensors="pt")
        return {
            'input_ids': inputs['input_ids'].squeeze(0),
            'attention_mask': inputs['attention_mask'].squeeze(0),
            'label': torch.tensor(self.labels[idx], dtype=torch.long)
        }

# Split dataset into training and testing sets
train_texts, test_texts, train_labels, test_labels =
train_test_split(df['Cleaned_Review'].values, df['Emotion'].values,
test_size=0.2, random_state=42)

# Create datasets
train_dataset = EmotionDataset(train_texts, train_labels)
test_dataset = EmotionDataset(test_texts, test_labels)

# Create DataLoaders
train_loader = DataLoader(train_dataset, batch_size=8, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=8, shuffle=False)

print("Dataset split completed!")
def train_model(model, train_loader, optimizer, criterion, device,
epochs=10):
    model.train()
    for epoch in range(epochs):
        total_loss, correct, total = 0, 0, 0
        for batch in train_loader:
            input_ids, attention_mask, labels =
batch['input_ids'].to(device), batch['attention_mask'].to(device),
batch['label'].to(device)

            optimizer.zero_grad()
```

```
            outputs = model(input_ids, attention_mask=attention_mask)
            loss = criterion(outputs.logits, labels)
            loss.backward()
            optimizer.step()
            scheduler.step()

            total_loss += loss.item()
            correct += (outputs.logits.argmax(dim=1) ==
labels).sum().item()
            total += labels.size(0)

        print(f"Epoch {epoch+1}: Loss={total_loss/len(train_loader):.4f},
Accuracy={correct/total:.4f}")
train_model(model, train_loader, optimizer, criterion, device, epochs=10)
# Set model to evaluation mode
model.eval()
all_preds, all_labels = [], []

with torch.no_grad():
    for batch in test_loader:
        input_ids, attention_mask, labels =
batch['input_ids'].to(device), batch['attention_mask'].to(device),
batch['label'].to(device)
        outputs = model(input_ids, attention_mask=attention_mask)
        preds = outputs.logits.argmax(dim=1)

        all_preds.extend(preds.cpu().numpy())
        all_labels.extend(labels.cpu().numpy())

# Print classification report
print("Classification Report:\n", classification_report(all_labels,
all_preds, target_names=label_encoder.classes_))


#HYBRID MODEL
import numpy as np
import pandas as pd
import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import Dataset, DataLoader
from transformers import BertTokenizer, BertModel
from sklearn.utils.class_weight import compute_class_weight
from torch.optim.lr_scheduler import CosineAnnealingLR

# Load Dataset
df = pd.read_csv("/kaggle/input/reviews-final/reviews_final.csv")
```

```
# Set device
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

# Encode Labels
emotion_labels = {label: idx for idx, label in
enumerate(df["Emotion"].unique())}
df["Label"] = df["Emotion"].map(emotion_labels)

# Compute class weights for imbalanced data
class_weights = compute_class_weight('balanced',
classes=np.unique(df["Label"]), y=df["Label"].values)
class_weights = torch.tensor(class_weights, dtype=torch.float).to(device)

# Custom Dataset
class ReviewDataset(Dataset):
    def __init__(self, texts, labels, tokenizer, max_len=128):
        self.texts = texts
        self.labels = labels
        self.tokenizer = tokenizer
        self.max_len = max_len

    def __len__(self):
        return len(self.texts)

    def __getitem__(self, idx):
        text = str(self.texts[idx])
        label = self.labels[idx]

        encoding = self.tokenizer(
            text,
            max_length=self.max_len,
            padding="max_length",
            truncation=True,
            return_tensors="pt"
        )

        return {
            "input_ids": encoding["input_ids"].squeeze(0),
            "attention_mask": encoding["attention_mask"].squeeze(0),
            "label": torch.tensor(label, dtype=torch.long)
        }

# Split Data
from sklearn.model_selection import train_test_split
texts = df["Cleaned_Review"].tolist()
labels = df["Label"].tolist()
```

```
train_texts, val_texts, train_labels, val_labels =
train_test_split(texts, labels, test_size=0.2, random_state=42,
stratify=labels)

# Create Dataloaders
train_dataset = ReviewDataset(train_texts, train_labels, tokenizer)
val_dataset = ReviewDataset(val_texts, val_labels, tokenizer)

train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=16, shuffle=False)
# Define Model
class BertBiLSTMAttention(nn.Module):
    def __init__(self, num_classes, hidden_dim=384, num_layers=2):
        super(BertBiLSTMAttention, self).__init__()
        self.bert = BertModel.from_pretrained("bert-base-uncased")
        self.layer_norm = nn.LayerNorm(768)  # Added Layer Normalization
        self.lstm = nn.LSTM(input_size=768, hidden_size=hidden_dim,
num_layers=num_layers,
                            bidirectional=True, batch_first=True,
dropout=0.3)
        self.attention = nn.Linear(hidden_dim * 2, 1)
        self.fc = nn.Linear(hidden_dim * 2, num_classes)
        self.dropout = nn.Dropout(0.5)

    def forward(self, input_ids, attention_mask):
        bert_output = self.bert(input_ids=input_ids,
attention_mask=attention_mask).last_hidden_state
        bert_output = self.layer_norm(bert_output)  # Normalization
before LSTM
        lstm_out, _ = self.lstm(bert_output)
        attention_weights =
torch.tanh(self.attention(lstm_out)).squeeze(2)
        attention_weights = torch.softmax(attention_weights, dim=1)
        context_vector = torch.sum(attention_weights.unsqueeze(2) *
lstm_out, dim=1)
        out = self.fc(self.dropout(context_vector))
        return out

# Initialize Model
num_classes = len(emotion_labels)
model = BertBiLSTMAttention(num_classes).to(device)
# Label Smoothing
class LabelSmoothingLoss(nn.Module):
    def __init__(self, classes, smoothing=0.1):
        super(LabelSmoothingLoss, self).__init__()
        self.confidence = 1.0 - smoothing
        self.smoothing = smoothing
        self.cls = classes
```

```
    def forward(self, pred, target):
        log_prob = torch.nn.functional.log_softmax(pred, dim=-1)
        target = torch.nn.functional.one_hot(target, self.cls).float()
        target = target * self.confidence + (1 - target) * self.smoothing
/ (self.cls - 1)
        return (-target * log_prob).sum(dim=-1).mean()

criterion = LabelSmoothingLoss(num_classes)  # Applied Label Smoothing
optimizer = optim.AdamW(model.parameters(), lr=2e-5, weight_decay=1e-4)
scheduler = CosineAnnealingLR(optimizer, T_max=10)  # Using Cosine
Annealing for LR scheduling

def train_model(model, train_loader, val_loader, criterion, optimizer,
scheduler, epochs=10):
    best_val_acc = 0
    for epoch in range(epochs):
        model.train()
        total_loss, total_correct = 0, 0
        for batch in train_loader:
            input_ids = batch["input_ids"].to(device)
            attention_mask = batch["attention_mask"].to(device)
            labels = batch["label"].to(device)

            optimizer.zero_grad()
            outputs = model(input_ids, attention_mask)
            loss = criterion(outputs, labels)
            loss.backward()
            optimizer.step()

            total_loss += loss.item()
            total_correct += (outputs.argmax(1) == labels).sum().item()
        scheduler.step()

        val_loss, val_correct = 0, 0
        model.eval()
        with torch.no_grad():
            for batch in val_loader:
                input_ids = batch["input_ids"].to(device)
                attention_mask = batch["attention_mask"].to(device)
                labels = batch["label"].to(device)
                outputs = model(input_ids, attention_mask)
                val_loss += criterion(outputs, labels).item()
                val_correct += (outputs.argmax(1) == labels).sum().item()

        val_acc = val_correct / len(val_dataset)
        print(f"Epoch {epoch+1}/{epochs}: Train Loss =
{total_loss/len(train_loader):.4f}, Train Acc =
```

```
{total_correct/len(train_dataset):.4f}, Val Loss =
{val_loss/len(val_loader):.4f}, Val Acc = {val_acc:.4f}")

        if val_acc > best_val_acc:
            best_val_acc = val_acc
            torch.save(model.state_dict(), "best_model.pth")

train_model(model, train_loader, val_loader, criterion, optimizer,
scheduler)
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, classification_report

def evaluate_model(model, val_loader, criterion):
    model.eval()
    val_loss, val_correct = 0, 0
    all_preds, all_labels = [], []

    with torch.no_grad():
        for batch in val_loader:
            input_ids = batch["input_ids"].to(device)
            attention_mask = batch["attention_mask"].to(device)
            labels = batch["label"].to(device)

            outputs = model(input_ids, attention_mask)
            loss = criterion(outputs, labels)
            val_loss += loss.item()
            val_correct += (outputs.argmax(1) == labels).sum().item()

            all_preds.extend(outputs.argmax(1).cpu().numpy())
            all_labels.extend(labels.cpu().numpy())

    val_acc = val_correct / len(val_dataset)
    print(f"Validation Loss: {val_loss / len(val_loader):.4f}, Validation
Accuracy: {val_acc:.4f}")

    # Generate confusion matrix
    cm = confusion_matrix(all_labels, all_preds)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap="Blues",
xticklabels=emotion_labels.keys(), yticklabels=emotion_labels.keys())
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.title("Confusion Matrix")
    plt.show()

    # Print classification report
```

```
    print("\nClassification Report:\n", classification_report(all_labels,
all_preds, target_names=emotion_labels.keys()))

# Run evaluation
evaluate_model(model, val_loader, criterion)
```