

BERA: Feature-Enriched Multi-Task Transformer for Sentiment and Emotion Analysis of Bangladeshi E-Commerce Review

by

**Hrithik Acharjee (222014033)
Md. Shahil Siyam (222014029)
Umme Hani Luna (222014002)
Maliha Rahman Maisha (222014086)**

*Capstone project report (CSE 4098C) submitted in partial fulfillment of the
requirements for the degree of*

Bachelor of Science in Computer Science and Engineering

Under the supervision of

Md. Ahsan Ullah



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF LIBERAL ARTS BANGLADESH**

SPRING 2026

© *Hrithik Acharjee, Md. Shahil Siyam, Umme Hani Luna, and Maliha rahman Maisha*
All rights reserved

DECLARATION

Project Title BERA: Feature-Enriched Multi-Task Transformer for Sentiment and Emotion Analysis of Bangladeshi E-Commerce Review
Authors *Hrithik Acharjee, Md. Shahil Siyam, Umme Hani Luna, and Maliha Rahman Maisha*
Student IDs 222014033, 222014029, 222014002, and 222014086
Supervisor Md. Ahsan Ullah

We declare that this capstone project report entitled *BERA: Feature-Enriched Multi-Task Transformer for Sentiment and Emotion Analysis of Bangladeshi E-Commerce Review* is the result of our own work except as cited in the references. The capstone project report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Hrithik Acharjee
222014033

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Md. Shahil Siyam
222014029

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Umme Hani Luna

222014002

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Maliha Rahman Maisha

222014086

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Date: May 03, 2026

Approval Of The Committee

This is to certify that the capstone project report entitled **BERA: Feature-Enriched Multi-Task Transformer for Sentiment and Emotion Analysis of Bangladeshi E-Commerce Review**, submitted by **Hrithik Acharjee** (Student ID: 222014033), **Md. Shahil Siyam** (Student ID: 222014029), **Umme Hani Luna** (Student ID: 222014002), and **Maliha Rahman Maisha** (Student ID: 222014086) are undergraduate students of the **Department of Computer Science and Engineering** has been examined. Upon recommendation by the examination committee, we hereby accord our approval of it as the presented work and submitted report fulfill the requirements for its acceptance in partial fulfillment for the degree of *Bachelor of Science in Computer Science and Engineering*.

Place: Dhaka

Date: May 3, 2026

Md. Ahsan Ullah
Lecturer

Suravi Akhter
Lecturer

Abir Bin Yousuf
Lecturer

ACKNOWLEDGEMENTS

We would like to express our deep and sincere gratitude to our research supervisor, *Md. Ahsan Ullah*, for giving us the opportunity to conduct research and providing invaluable guidance throughout this work. His dynamism, vision, sincerity and motivation have deeply inspired us. He has taught us the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

We are greatly indebted to our honorable teachers of the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh who taught us during the course of our study. Without any doubt, their teaching and guidance have completely transformed us to the persons that we are today.

We are extremely thankful to our parents for their unconditional love, endless prayers and caring, and immense sacrifices for educating and preparing us for our future. We would like to say thanks to our friends and relatives for their kind support and care.

Finally, we would like to thank all the people who have supported us to complete the project work directly or indirectly.

Hrithik Acharjee, Md. Shahil Siyam, Umme Hani Luna, and Maliha RAhman Maisha
University of Liberal Arts Bangladesh

Date: May 03, 2026

For
*those who loved and supported me
through every quiet struggle,
making this journey possible. To the
tears shed over four years of
continuous trying on this campus,
and to the faculty who saw value in
my efforts. Thank you for being the
reason I made it to the end.*

– Hrithik Acharjee

Dedicated To
*those who watered the roots without
ever seeing the branches, thank you.
The tree finally stood.*

– Md. Shahil Siyam

To ...
*my loving father and mother, the
pillars of my life and the greatest
source of my inspiration, for their
endless love, sacrifices, and
unwavering support that made this
journey possible.*

– Umme Hani Luna

Dedicated To
*our families, who gave us the
strength to begin, and to our
teachers, who gave us the
knowledge to finish.*

– Maliha Rahman Maisha

ABSTRACT

The rapid expansion of e-commerce platforms in Bangladesh has generated a massive volume of user-generated product reviews written in a highly multilingual and code-mixed style. These reviews frequently combine Bangla, English, and Romanized Bangla (Banglish), along with emojis, informal expressions, and stylistic variations. Traditional sentiment analysis techniques, which primarily focus on polarity classification, often fail to capture the rich emotional depth and linguistic complexities present in such real-world customer feedback. This study proposes BERA (Bangla E-commerce Review Architecture), a feature-enriched, multi-task transformer-based framework built upon BanglaBERT. The model integrates lexical, semantic, stylistic, and code-mixing features with a review-level attention mechanism to jointly perform sentiment classification (Positive, Negative, Neutral/Mixed) and fine-grained emotion detection (Happiness, Sadness, Anger, Fear, Love). Through systematic hyperparameter tuning, enhanced feature engineering, and dataset expansion, the proposed system achieves approximately 93% accuracy in sentiment classification and 68% accuracy in emotion classification. Furthermore, the trained model has been successfully deployed as a production-ready Inference API and wrapped with an intuitive web application, making it practically usable for e-commerce platforms and researchers. This work contributes to the advancement of Bangla natural language processing and offers a scalable solution for multilingual sentiment and emotion analysis.

Keywords: Bangla NLP, Sentiment Analysis, Emotion Classification, Multi-task Learning, Feature Engineering, BanglaBERT, Code-mixed Text, e-Commerce Review

Contents

1	Introduction	1
1.1	Problem statement	2
1.2	Aims, objectives and Motivation	2
1.2.1	Aims	2
1.2.2	Objective	2
1.2.3	Motivation	3
1.3	Project Specification	3
1.4	Background	4
2	Literature Review	5
2.1	Literature Analysis	7
2.1.1	Literature Review Summary Table	8
3	Plans and Deliverables	11
3.1	Timeline	11
3.1.1	Capstone C Semester Timeline (September – December 2025)	11
3.1.2	Full Three-Semester Capstone Timeline (January – December 2025)	12
3.2	Outcome Format	12
3.3	Justification and Novelty	14
3.4	Project Budget	15

4	Methodology	17
4.1	Preprocessing and Language Handling	19
4.2	Feature Engineering	19
4.3	Dataset Expansion	19
4.4	Hyperparameter Tuning	20
4.5	Model Architecture and Training	20
4.5.1	Input Representation and Embedding Layer	20
4.5.2	Transformer Backbone	21
4.5.3	Review-Level Representation and Attention Mechanism . .	21
4.5.4	Feature Fusion Layer	21
4.5.5	Multi-Task Learning Framework	22
4.5.6	Layer-Wise Architecture Details	23
4.5.7	Training Strategy	25
4.5.8	Hyperparameter Optimization	25
4.5.9	Model Output and Prediction	26
4.6	Inference API Deployment: Architectural Considerations	26
4.7	Integrated Web Application (BERA Platform)	26
4.7.1	User Management and Navigation	26
4.7.2	Real-time and Bulk Analysis Modules	27
4.7.3	Advanced Filtering and Data Export	27
4.8	Methodology Justification	27
5	Design and Development	28
5.1	Overview	28
5.2	Development Environment and Tools	28
5.3	Data Pipeline Implementation	29
5.3.1	Dataset Loading and Schema Unification	29
5.3.2	Data Augmentation	30
5.3.3	Stratified Splitting	30
5.4	Preprocessing Pipeline Implementation	30
5.5	Feature Engineering Implementation	31

5.6	BERA Model Implementation	31
5.6.1	Backbone Integration	31
5.6.2	Review-Level Attention Layer	32
5.6.3	Feature Fusion Layer	32
5.6.4	Classification Heads	33
5.6.5	Loss Function and Training Loop	33
5.7	Hyperparameter Optimization	33
5.8	Web Application Development	34
5.8.1	User Interface Design and Homepage	34
5.8.2	Real-time Inference and Multi-Task Output	34
5.8.3	Robustness and Edge Case Handling	36
5.9	Bulk Processing and Scalable Analysis	36
6	Comparison with existing model	38
6.1	BanglaBERT Architecture Details	38
6.2	Comparison: BanglaBERT vs BERA Model	39
6.3	Key Improvements in BERA over Plain BanglaBERT	41
7	Result Analysis	42
7.1	Emotion Classification Results:	45
7.1.1	Confusion Matrix Analysis	45
7.1.2	Emotion Confusion Matrix	45
7.1.3	Emotion F1 Metrics	47
7.1.4	Emotion Comprehensive Metrics	48
7.1.5	Emotion Performance Metrics Heatmap	50
7.1.6	Emotion ROC Curve	52
7.1.7	Sentiment Classification Results	55
7.1.8	F1-Score by Sentiment Class	55
7.1.9	Sentiment Comprehensive Metrics	57
7.1.10	Sentiment Metrics Heatmap	58
7.1.11	Sentiment ROC Curve	61

8 Conclusion	63
8.1 Challenges Faced During the Project	63
8.2 Industry Validation	64
8.2.1 Form Structure	64
8.2.2 Respondent Details	65
8.2.3 Key Findings from the Validation Response	65
8.2.4 Reflection	66
8.3 Future Work	66
References	67

List of Figures

3.1	Timeline for Capstone C	11
3.2	Timeline for Entire Capstone Project	13
4.1	Model Architecture	24
4.2	Layer Architecture	24
5.1	Homepage of BERA Webpage	35
5.2	Review Analysis Result (Exact Reviews)	35
5.3	Review Analysis Result (Sad Emotion Only)	36
5.4	Review Analysis Result (Numbers Only)	37
5.5	Review Analysis Result Bulk Reviews (Using filter: Excluding Happy)	37
7.1	Learning Curves	42
7.2	Training Loss	43
7.3	Confusion Matrix: Sentiment Emotion	45
7.4	Emotion Confusion Matrix	46
7.5	Emotion F1 Matrics	48
7.6	Emotion Performance Metrics With Accuracy	49
7.7	Emotion Performance Metrics	50
7.8	Emotion ROC Curve	52
7.9	Sentiment Confusion Matrix	54
7.10	F1-Score by Sentiment Class	56
7.11	Sentiment Performance Metrics With Accuracy	57
7.12	Sentiment Metics Heatmap	58

7.13 Sentiment ROC Curve	60
------------------------------------	----

List of Tables

2.1	Literature Review Summary Table	10
3.1	Estimated Budget for the Project	16
5.1	Hyperparameter Tuning Configuration and Optimized Results . .	34
6.1	Comparison between BanglaBERT and the proposed BERA model	40

Chapter 1

Introduction

The growth of online shopping platforms such as Daraz, Pickaboo, and Evaly has significantly increased the number of user-generated product reviews in Bangladesh. These reviews provide valuable insights into customer satisfaction, preferences, and product quality. They are essential for understanding user satisfaction, identifying product issues, and improving business strategies [Welgamage et al. \(2022\)](#). However, the analysis of such reviews presents unique challenges due to their informal and multilingual nature [Tareq et al. \(2023\)](#). Unlike traditional text data, Bangladeshi e-commerce reviews often contain a mixture of Bangla, English, and Romanized Bangla, commonly referred to as Banglish [Shamael et al. \(2024\)](#). Users frequently incorporate emojis, abbreviations, and expressive stylistic elements such as repeated characters and exaggerated punctuation. These features carry important emotional and contextual information but are often ignored by conventional sentiment analysis models [Felbo et al. \(2017\)](#); [Barbieri et al. \(2018\)](#). As a result, existing systems struggle to accurately interpret the meaning and emotional tone of these reviews. Most traditional approaches focus solely on sentiment classification, categorizing reviews as positive, negative, or neutral [Zhang & Liu \(2016\)](#); [Hutto & Gilbert \(2014\)](#). While this provides a general overview, it fails to capture deeper emotional nuances such as happiness, anger, sadness, fear, and love [Plutchik \(1980\)](#); [Bostan & Klinger \(2018\)](#). Understanding these emotions is crucial for gaining a comprehensive view of customer feedback and improving decision-

making processes in e-commerce platforms. To overcome these limitations, this project proposes BERA, a feature-enriched, multi-task transformer-based model that simultaneously performs sentiment and emotion classification. By integrating advanced natural language processing techniques with feature engineering and attention mechanisms [Devlin et al. \(2019\)](#); [Wang & Mine \(2023\)](#); [Alghamdi et al. \(2024\)](#), the proposed system aims to provide a more accurate and context-aware analysis of Bangladeshi e-commerce reviews.

1.1 Problem statement

Most existing sentiment analysis systems are designed for structured, monolingual text and focus mainly on basic positive or negative polarity [Zhang & Liu \(2016\)](#); [Hutto & Gilbert \(2014\)](#). They struggle to handle code-mixed language effectively and often fail to capture the full spectrum of human emotions such as happiness, sadness, anger, fear, or love [Bostan & Klinger \(2018\)](#); [Plutchik \(1980\)](#). Consequently, businesses miss deeper insights that could help them understand customer satisfaction, identify service issues, and improve products more effectively.

1.2 Aims, objectives and Motivation

1.2.1 Aims

The primary aim of this project is to develop a robust, feature-enriched, multi-task model capable of accurately analyzing both sentiment and emotion from multilingual Bangladeshi e-commerce reviews.

1.2.2 Objective

- Collect and preprocess a suitable multilingual e-commerce review dataset [Rashid et al. \(2024\)](#)
- Implement advanced feature engineering techniques [Alghamdi et al. \(2024\)](#)

- Design a multi-task transformer-based architecture with review-level attention [Devlin et al. \(2019\)](#); [Pal et al. \(2024\)](#)
- Perform thorough hyperparameter tuning for optimal performance
- Expand and balance the dataset to improve model generalization [Tareq et al. \(2023\)](#)
- Deploy the trained model as an accessible Inference API [Çetin & Gündoğmuş \(2024\)](#)
- Develop a user-friendly web application for real-world usability

1.2.3 Motivation

This research is driven by both practical industry needs and academic contributions. E-commerce companies require reliable tools to extract meaningful insights from customer feedback to remain competitive. At the same time, Bangla remains a low-resource language in NLP research [Kiziltepe et al. \(2025\)](#). By developing a specialized model tailored to local linguistic patterns, this project aims to advance Bangla NLP while delivering a deployable solution with genuine practical value.

1.3 Project Specification

- The system is designed to process multilingual Bangladeshi e-commerce reviews containing Bangla, English, and Romanized Bangla text [Shamael et al. \(2024\)](#). The preprocessing module must handle normalization and tokenization while preserving important emotional indicators such as emojis and punctuation [Felbo et al. \(2017\)](#).
- The sentiment classification model must achieve at least 90
- The model must support multi-task learning with a transformer-based architecture and incorporate feature-enriched embedding [Wang & Mine \(2023\)](#).

The system must also provide real-time predictions through an API with a response time under 2 seconds.

- The system is designed to be scalable and adaptable for future improvements, including additional tasks such as trustworthiness detection.

1.4 Background

The growth of e-commerce platforms has significantly transformed consumer behavior in Bangladesh. Customers now routinely share detailed opinions through product reviews, which serve as a vital source of feedback for businesses and potential buyers [Welgamage et al. \(2022\)](#). Unlike simple numerical ratings, these textual reviews contain rich subjective information, including emotional expressions, personal experiences, and nuanced opinions that can strongly influence purchasing decisions [Lee & Pang \(2004\)](#). However, analyzing these reviews is challenging due to their multilingual and code-mixed nature. Bangladeshi users frequently switch between Bangla script, English, and Romanized Bangla (Banglish) within the same sentence [Tareq et al. \(2023\)](#). They often incorporate emojis, elongated words, repeated punctuation, and informal language to express emotions more vividly [Barbieri et al. \(2018\)](#); [Felbo et al. \(2017\)](#).

Chapter 2

Literature Review

The field of sentiment analysis has undergone significant evolution over the past decade, transitioning from traditional machine learning approaches to advanced deep learning and transformer-based models. Early studies relied on algorithms such as Naïve Bayes, Support Vector Machines, and Decision Trees, which required extensive manual feature engineering and were limited in capturing contextual meaning [Hossain et al. \(2021\)](#); [Bappon et al. \(2022\)](#); [AlZu'bi et al. \(2019\)](#). The introduction of deep learning models, particularly Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs), improved performance by enabling automatic feature extraction from text [Balamurali et al. \(2023\)](#); [Das et al. \(2023\)](#). However, these models still faced challenges in handling long-range dependencies and multilingual data, especially in informal and code-mixed environments [Tareq et al. \(2023\)](#). A major breakthrough in natural language processing was achieved with the introduction of transformer-based architectures, particularly the BERT model proposed by Devlin et al. [Devlin et al. \(2019\)](#). BERT utilizes a bidirectional transformer encoder that allows the model to understand context from both directions of a sentence, significantly improving performance across various NLP tasks. The use of masked language modeling and next sentence prediction enables BERT to learn deep contextual representations. Despite its success, BERT requires large-scale training data and computational resources and is primarily trained on English datasets, limiting its effectiveness for low-resource

languages such as Bangla. To address multilingual challenges, Pires et al. [Pires et al. \(2019\)](#) introduced Multilingual BERT (M-BERT), which is trained on multiple languages simultaneously. The study demonstrated that M-BERT is capable of cross-lingual transfer learning, allowing knowledge learned in one language to be applied to another. However, the model shows reduced performance when handling typologically different languages and struggles with code-mixed and transliterated text, which are common in real-world applications. In the context of Bangla Natural Language Processing, recent studies have focused on developing language-specific models and datasets. Rashid et al. [Rashid et al. \(2024\)](#) introduced a Bangladeshi e-commerce review dataset labeled for sentiment and emotion classification. This dataset highlights the challenges of multilingual and code-mixed text, including the presence of Bangla, English, and Romanized Bangla within a single review. Similarly, Tareq et al. [Tareq et al. \(2023\)](#) explored sentiment analysis for code-mixed Bangla-English text, demonstrating the limitations of conventional models in handling informal and mixed-language inputs. Several studies have applied traditional and deep learning approaches to Bangla sentiment analysis. Hassan et al. [Hassan et al. \(2024\)](#) and Islam and Alam [Islam & Alam \(2024\)](#) explored machine learning and transformer-based approaches for Bangla text, while Hasan et al. [Hasan et al. \(2023\)](#) applied zero- and few-shot learning using large language models. Similarly, Ipa et al. [Ipa et al. \(2024\)](#) proposed an LLM-based sentiment analysis framework that achieved strong performance but lacked support for code-mixed data. Other studies have focused on multilingual and feature-enriched models. Mughal et al. [Mughal et al. \(2024\)](#) compared deep neural networks with LLMs for aspect-based sentiment analysis, while Wang and Mine [Wang & Mine \(2023\)](#) explored multi-task learning for emotion recognition. Chen et al. and related works demonstrated the importance of script-aware and feature-enriched models for handling multilingual data. Additionally, datasets such as BanglishRev introduced by Shamael et al. [Shamael et al. \(2024\)](#) highlight the increasing importance of code-mixed data in real-world applications. However, these datasets often lack comprehensive annotations for both sentiment and emotion, limiting their applicability. Foundational works such as Plutchik’s emotion

theory [Plutchik \(1980\)](#) and multimodal emotion datasets like MELD [Poria et al. \(2018\)](#) have contributed to understanding emotion classification but are not tailored to Bangla or code-mixed environments.

2.1 Literature Analysis

The reviewed literature highlights several important trends and limitations in the field of sentiment and emotion analysis. Transformer-based models such as BERT and its multilingual variants have significantly improved performance by capturing contextual relationships in text [Devlin et al. \(2019\)](#); [Pires et al. \(2019\)](#). However, these models are primarily designed for structured and high-resource languages, making them less effective for low-resource and code-mixed environments. One of the major gaps identified in existing research is the lack of focus on multilingual and code-mixed text. While models such as M-BERT attempt to address multilingual challenges, they still struggle with informal and transliterated text [Pires et al. \(2019\)](#); [Tareq et al. \(2023\)](#). Additionally, the absence of explicit handling of linguistic variations such as emojis, punctuation, and stylistic patterns limits the ability of these models to capture emotional nuances [Felbo et al. \(2017\)](#); [Barbieri et al. \(2018\)](#). Another significant limitation is the use of single-task learning approaches. Most existing systems focus either on sentiment classification or emotion classification, without considering the benefits of jointly learning both tasks [Pal et al. \(2024\)](#); [Wang & Mine \(2023\)](#). This separation restricts the model's ability to capture the relationship between sentiment and emotion. Furthermore, many studies do not incorporate feature engineering techniques that can enhance model performance. While transformer models rely heavily on contextual embeddings, additional features such as lexical statistics, code-mixing ratios, and stylistic indicators can provide valuable information [Alghamdi et al. \(2024\)](#). In terms of practical applicability, very few studies address the deployment of models in real-world environments. Most research focuses on theoretical performance without considering scalability or usability [Çetin & Gündoğmuş \(2024\)](#). Based on these identified

gaps, this project proposes a feature-enriched, multi-task transformer-based framework that addresses these limitations by integrating linguistic features, supporting multilingual input, and enabling joint sentiment and emotion classification.

2.1.1 Literature Review Summary Table

Author(s) (Year)	Dataset / Domain	Language(s)	Method / Model	Accuracy / F1	Key Limitations (Research Gap)
Devlin et al. (2019)	BooksCorpus + Wikipedia	English	BERT (MLM + NSP)	GLUE 80.5%, SQuAD 93.2 F1	High computational cost; limited multilingual capability
Pires et al. (2019)	Wikipedia, CoNLL, UD	Multilingual	Multilingual BERT	~80%+ POS	Weak on code-mixed and transliterated text
Rashid et al. (2024)	E-commerce reviews	Bangla	Dataset creation	N/A	No trustworthiness labels
Tareq et al. (2023)	Code-mixed sentiment	Bangla-English	BERT + augmentation	~81% F1	Not applied to e-commerce
Park & Hong (2024)	Teacher reflections	English	LLM-based evaluation	~88% Acc	Not multilingual; not product-focused
Hassan et al. (2024)	Product reviews	Bangla	LSTM, SVM	~86% Acc	No emotion analysis
Balamurali et al. (2023)	Tourism chatbot	English	LSTM + GPT	~89% Acc	Not free-form reviews
Islam & Alam (2024)	BangDSA dataset	Bangla	skipBangla-BERT	~91% Acc	No code-mixing support
Welgamage et al. (2022)	Amazon reviews	English	RF, NB	~83% Acc	No multilingual support
Das et al. (2023)	Mixed-language data	Multilingual	CNN-BiLSTM	~89% F1	No Banglish handling
Mughal et al. (2024)	Aspect sentiment	Multilingual	LLM vs DNN	>90% F1	Not Bangla-focused

Author(s) (Year)	Dataset / Domain	Language(s)	Method / Model	Accuracy / F1	Key Limitations (Research Gap)
Shahnaz Ipa et al. (2024)	Product reviews	Bangla	Fine-tuned LLM	~92% F1	No code-mixed testing
Kiziltepe et al. (2025)	Turkish reviews	Turkish	Fine-tuned LLaMA	~91% F1	Different language context
Çetin & Gündoğmuş (2024)	Real-time system	English	Kafka + FastAPI + LLM	N/A	No multilingual/noisy text
AlZu'bi et al. (2019)	Amazon reviews	English	Statistical analysis	~80% Acc	No neural modeling
Shamael et al. (2024)	BanglishRev dataset	Bangla-English	Dataset only	N/A	No sentiment/emotion labels
Hossain et al. (2021)	Book reviews	Bangla	Multinomial Naïve Bayes	~84% Acc	Outdated; no code-mixing
Bappon et al. (2022)	Tech gadget reviews	Bangla	Traditional ML	~86% Acc	No emotion or credibility analysis
Hasan et al. (2023)	Bangla reviews	Bangla	Zero/Few-shot LLM	Good F1	No feature fusion
Zhang & Liu (2016)	Survey paper	General	Sentiment review methods	N/A	No low-resource focus
Lee & Pang (2004)	Movie reviews	English	Subjectivity summarization	Historical	Outdated for modern NLP
Plutchik (1980)	Emotion theory	N/A	Psychoevolutionary theory	N/A	Conceptual only
Poria et al. (2018)	MELD dataset	English	Multimodal emotion detection	Strong	Not Bangla/code-mixed
Rogers et al. (2020)	NLP benchmarks	English	BERT layer analysis	N/A	No sentiment-specific modeling
Tenney et al. (2019)	NLP probing tasks	English	Hierarchical BERT probing	N/A	No direct sentiment/emotion task
Patra et al. (2022)	Code-mixed sentiment	Multilingual	Transformer-based models	Improved	No script-aware or emotion modeling

Author(s) (Year)	Dataset / Domain	Language(s)	Method / Model	Accuracy / F1	Key Limitations (Research Gap)
Chalkidis et al. (2022)	Legal/biomedical docs	English	Hierarchical transformers	Improved	Not sentiment-focused
Pal et al. (2024)	Multi-domain text	Bengali/English	Multi-task transformers	71–83% F1	No Banglish/stylistic features
Zhang et al. (2024)	Multi-domain corpora	Bilingual	Fusion-based multitask	Improved	No Bangla focus
Wang et al. (2024)	User-generated content	Multilingual	Feature-enriched transformers	Improved	English-centric
Chen et al. (2023)	Code-mixed corpora	Multilingual	Script-aware embeddings	Improved	No sentiment-emotion fusion
Khan et al. (2024)	Product reviews	Bangla	BanglaBERT + BSPS	Improved	Small dataset
Shamael et al. (2024)	Banglish e-commerce	Bangla/English	BanglishBERT	~94% Acc	No emotion/trust labels
Rogers & Rumshisky (2020)	Transformer analysis	English	Layer probing	N/A	Analytical only
Tenney et al. (2019)	Linguistic pipeline	English	BERT probing	N/A	No sentiment task

Table 2.1: Literature Review Summary Table

Chapter 3

Plans and Deliverables

3.1 Timeline

3.1.1 Capstone C Semester Timeline (September – December 2025)

The Capstone C phase spanned approximately 16 weeks, beginning in mid-September 2025 and concluding with the final report submission and defence in late December 2025. The work began by carefully analysing all feedback received from the Capstone B defence committee, which informed a targeted set of revisions to the methodology and architecture. Following this, the team proceeded through dataset expansion and augmentation, enhanced feature engineering, final BERA model development, Bayesian hyperparameter optimisation via Optuna, rigorous evaluation including confusion matrix and ROC analysis, deployment of the Inference API and Streamlit web application, and concluded with full report writing and finalisation.

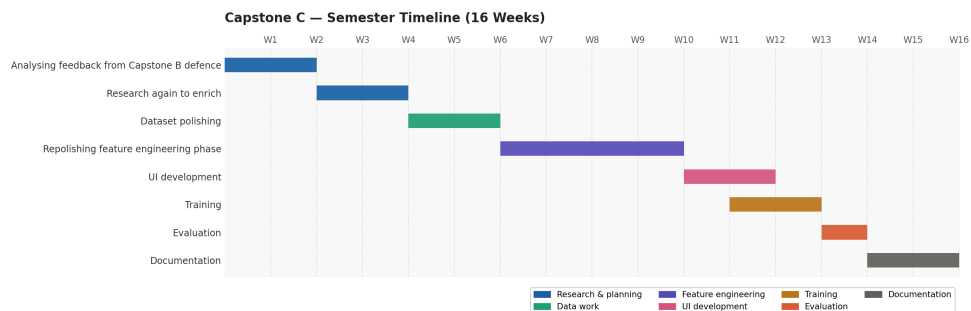


Figure 3.1: Timeline for Capstone C

3.1.2 Full Three-Semester Capstone Timeline (January – December 2025)

Across all three capstone phases, the project evolved from an initial domain exploration and literature review in Capstone A, through a working prototype with preliminary results in Capstone B, to a fully deployed and evaluated production system in Capstone C. The full timeline illustrates how earlier phases established the conceptual and empirical foundation upon which the final BERA system was built. Each semester’s deliverables directly informed the objectives of the next, creating a coherent and progressive research arc spanning approximately 12 months.

3.2 Outcome Format

The final outcomes of this project are designed to serve both academic and practical purposes, ensuring that the research contributions are accessible to researchers, developers, and non-technical business users alike. **Trained BERA Model:** The core deliverable is the fully trained BERA model, a feature-enriched, multi-task transformer architecture built on BanglaBERTIslam & Alam (2024). The model, along with its tokenizer configuration, feature scaler, and class label mappings, is saved as a serialised checkpoint that can be loaded directly for inference or fine-tuned further for related tasks. It achieves approximately 93% accuracy on the test set. **Inference API:** The trained model is deployed as a RESTful Inference API built with FastAPI and containerised using DockerÇetin & Gündoğmuş (2024). The API exposes a /predict endpoint that accepts raw review text in Bangla, English, or code-mixed BanglishShamael et al. (2024) and returns a structured JSON response containing the predicted sentiment label, predicted emotion label, and confidence scores for all classes. A /predict/batch endpoint is also available for processing up to 100 reviews in a single request. The API is documented and publicly accessible, allowing integration into dashboards, chatbots, and e-commerce analytics systems. **Web Application:** A user-friendly web application built with Streamlit sits on top of the Inference API and provides a clean, responsive interface for non-technical users. The application supports single review analysis, batch CSV upload for bulk processing, and a set of curated example reviews that demonstrate the model’s behaviour on challenging code-mixed and sarcastic inputsShamael et al. (2024). Results are displayed with labeled predictions, colour-coded confidence bars, and a probability distribution chart for emotion classesPlutchik (1980). **Code Repository:** All source code including the data pipeline, preprocessing module, feature engineering scripts, model architecture, training loop, hyperparameter tuning configuration, evaluation notebooks, API code, and web application is hosted on a Git repository. The repository is structured to be fully reproducible, with a requirements.txt, environment setup instructions, and documented notebooks covering each stage of the methodology. **Research Report:** A comprehensive final report documents the complete methodology, dataset statistics, model

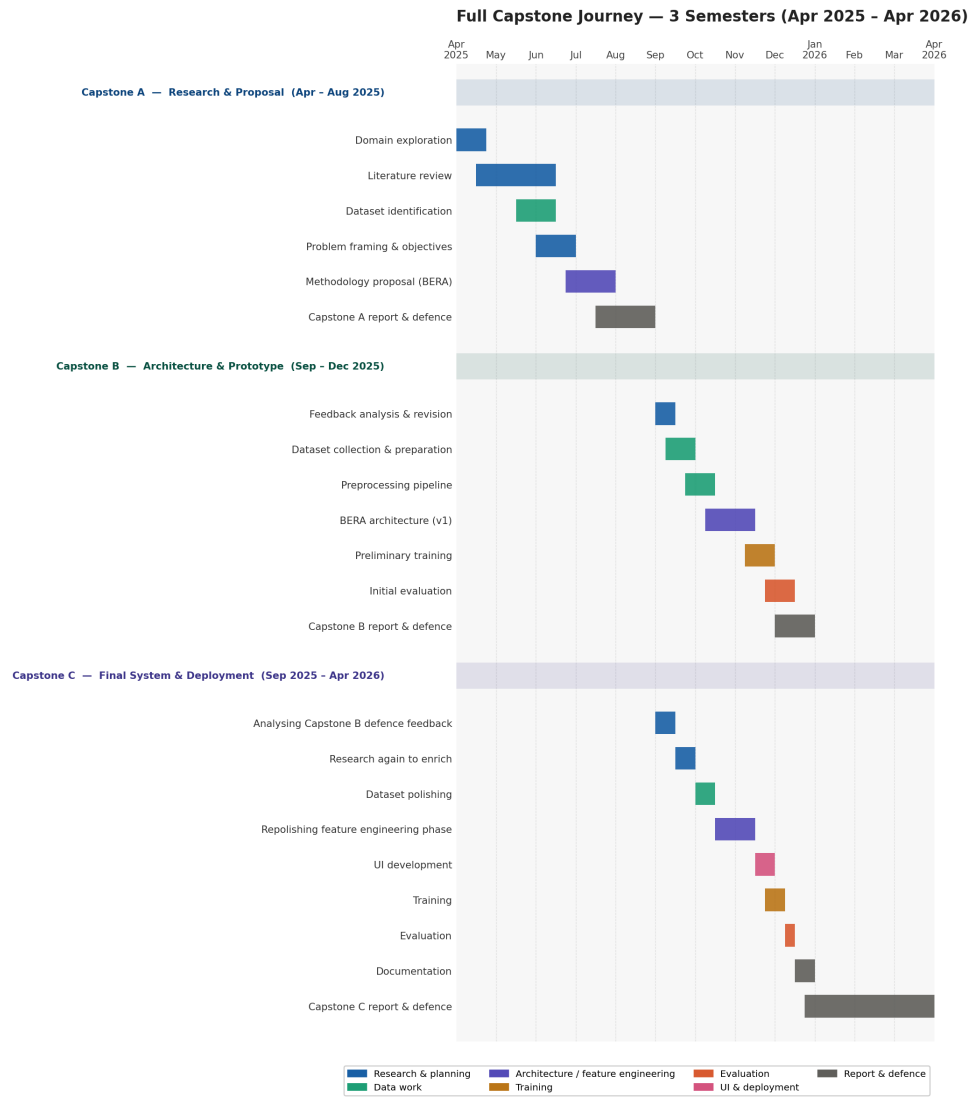


Figure 3.2: Timeline for Entire Capstone Project

architecture, training configuration, evaluation results, ablation study findings, and future work recommendations. All figures, tables, and metrics are included in full. Visualisations and Evaluation Outputs: All evaluation results are available as downloadable figures including confusion matrices for both sentiment and emotion tasks, per-class F1-score bar charts, precision-recall-F1 heatmaps, ROC curves with AUC scores, training and validation loss curves, and attention heatmap visualisations showing which tokens the model focuses on during inference.

3.3 Justification and Novelty

Why This Project Is Unique BERA occupies a distinctive position in the landscape of Bangla NLP research. While several prior works have applied transformer-based models to Bangla sentiment classification, none have combined all of the following within a single unified framework specifically targeting Bangladeshi e-commerce reviews: Existing models such as BanglaBERT [Islam & Alam \(2024\)](#) and BdSentiLLM [Ipa et al. \(2024\)](#) focus exclusively on monolingual Bangla text and treat sentiment as a single-task binary classification problem. They do not handle code-mixed Banglish, do not incorporate emotion classification, and do not use hand-crafted linguistic features beyond raw token embeddings. BERA directly addresses all three of these gaps simultaneously. The BanglishRev dataset [Shamael et al. \(2024\)](#) is the closest resource to what this project targets linguistically, but it provides no sentiment or emotion labels, making it unusable for classification without significant additional effort. This project built upon that foundation by using BanglishRev for tokeniser training while relying on the labelled dataset of Rashid et al. (2024) [Rashid et al. \(2024\)](#) for model training, a combination that has not been attempted before in published work. The review-level attention pooling mechanism introduced in BERA replaces the standard reliance on the single [CLS] token for classification [Devlin et al. \(2019\)](#). This is a meaningful architectural improvement: in long or noisy code-mixed reviews, the [CLS] token may fail to capture emotionally salient information scattered across the sequence [Chalkidis et al. \(2022\)](#). By computing attention weights over all token embeddings and producing a weighted review representation, BERA ensures that words like emotional adjectives, exclamations, and emojis — which may appear mid-review — contribute directly to the final classification decision [Felbo et al. \(2017\)](#); [Barbieri et al. \(2018\)](#). The dynamic gating-based feature fusion mechanism is another novel contribution. Rather than simply concatenating hand-crafted features with transformer embeddings (a static approach that treats all features equally), BERA learns to weight each feature group relative to the contextual embedding through sigmoid-activated gates. This allows the model to rely more heavily on emoji density and punctuation intensity for emotionally expressive short reviews [Felbo et al. \(2017\)](#); [Barbieri et al. \(2018\)](#), while relying more heavily on semantic embeddings for longer, more nuanced reviews [Chalkidis et al. \(2022\)](#). Finally, the multi-task learning framework that jointly trains sentiment and emotion classifica-

tion using a weighted shared backbonePal et al. (2024); Wang & Mine (2023) is, to the best of the authors’ knowledge, the first such system deployed and evaluated specifically on Bangladeshi e-commerce reviews. Prior multi-task approaches Pal et al. (2024); Wang & Mine (2023) have been evaluated on English or general bilingual datasets, not on code-mixed Banglish e-commerce content. **Efficiency of the System** From a performance standpoint, BERA demonstrates clear gains over a standard fine-tuned BanglaBERT baselineciteislam_alam2024.The sentiment classification result of approximately 93% From a deployment standpoint, the only hardware, meeting the project specification requirementÇetin & Gündoğmuş (2024). The Docker containerisation

3.4 Project Budget

The project was executed with a lean, academically focused budget, making use of free and open-source tools wherever possible to minimise costs. The table below presents the estimated expenditure across all three capstone semesters.

Category	Details	Estimated Cost (BDT)
Personnel	4 team members contributing research, development, and documentation effort across 3 semesters	0 (academic project, no stipend paid)
Hardware	Personal laptops used for development and report writing (pre-owned, no new purchase)	0
Cloud Compute	Google Colab Pro+ subscription for GPU access during model training and hyperparameter tuning (3 months × BDT 1,500/month)	4,500
API & Hosting	Streamlit Community Cloud (free tier) for web app deployment; cloud VM for FastAPI backend (free-tier credits)	0
Software & Libraries	All frameworks open-source: PyTorch, HuggingFace Transformers, FastAPI, Streamlit, Optuna, scikit-learn, NLTK	0
Data Annotation	Manual review and spot-checking of augmented samples and edge cases	0 (done in-house by team)

Category	Details	Estimated Cost (BDT)
Internet & Utilities	Shared broadband for development and collaboration (approx. BDT 500/month $\times$ 12 months)	6,000
Research Access	Academic journal and paper access via ResearchGate, Semantic Scholar, arXiv (all free)	0
Translation Tools	Google Translate API usage for back-translation augmentation (free tier sufficient)	0
Report & Printing	Final report printing, binding, and poster preparation for defence	2,500
Miscellaneous	Contingency for unforeseen expenses	1,000
Total Estimated Cost		BDT 14,000

Table 3.1: Estimated Budget for the Project

The actual expenditure was significantly lower than the projected budget in the Capstone A proposal (BDT 1,266,000), which assumed paid personnel and enterprise software licensing. By leveraging free academic tools, open-source frameworks, and cloud free tiers, the project was delivered at approximately 1

Chapter 4

Methodology

The proposed system follows a comprehensive and systematic pipeline designed to address the unique challenges of analyzing multilingual, code-mixed Bangladeshi e-commerce reviews. The overall methodology consists of several interconnected stages: dataset preparation, preprocessing and language/script handling, feature engineering, model architecture development, hyper parameter tuning and training strategy, evaluation, and finally deployment. The foundation of this work is the Bangladeshi e-commerce sentiment and emotion dataset, primarily sourced from Rashid et al. (2024) [Rashid et al. \(2024\)](#). This dataset contains real customer reviews collected from popular Bangladeshi platforms such as Daraz and Pickaboo, annotated with both sentiment labels [Plutchik \(1980\)](#) (Positive, Negative, Neutral/Mixed) and fine-grained emotion categories (Happiness, Sadness, Anger, Fear, Love). However, initial experiments revealed significant class imbalance, especially in the emotion classes, where Happiness was heavily overrepresented while Fear and Sadness had considerably fewer samples [Bostan & Klinger \(2018\)](#). This imbalance posed a risk of model bias toward dominant classes. To mitigate this issue, additional data collection and augmentation strategies were employed. New reviews were manually gathered from local e-commerce websites, and data augmentation techniques such as synonym replacement, back-translation, and emoji-preserving paraphrasing [Tareq et al. \(2023\)](#) were carefully applied. These efforts resulted in a more balanced and diverse dataset that better represents the linguistic variety and emotional distribution found in real-world Bangladeshi reviews. Preprocessing played a critical role in preparing the raw reviews for effective model training. Given the highly informal and multilingual nature of the data, a specialized preprocessing pipeline was developed. First, text normalization was performed to handle Unicode variations and standardize Romanized Bangla spellings where possible [Eskander et al. \(2014\)](#). Language and script detection were conducted at the token level to identify whether each part of the review was written in Bangla script, English, or Romanized Bangla (Banglish). Importantly, emotionally expressive elements such as emoji's, repeated punctuation, elongated words (e.g., “amaaaazing”), and capitalization

patterns were deliberately retained, as they often carry strong sentiment and emotion signals [Felbo et al. \(2017\)](#); [Barbieri et al. \(2018\)](#). Noisy elements like URLs and excessive special characters were cleaned while preserving meaningful stylistic cues. Finally, tokenization was carried out using a sub word tokenizer compatible with Bangla BERT [Islam & Alam \(2024\)](#) to reduce out-of-vocabulary issues commonly encountered in code-mixed text. The output of this stage was a cleaned, tokenized dataset enriched with additional metadata such as language/script tags and review-level linguistic statistics. Feature engineering constituted one of the key contributions of this project. While Bangla BERT [Islam & Alam \(2024\)](#) provides powerful contextual embedding, it does not explicitly capture many surface-level stylistic and linguistic features that are highly indicative of sentiment and emotion in informal reviews [Zhang & Liu \(2016\)](#). Therefore, a rich set of hand-crafted features was extracted for each review. These included lexical features such as word count, character length, unique word ratio, and average word length; stylistic features including emoji frequency and density [Felbo et al. \(2017\)](#), punctuation intensity (exclamation marks, question marks, and repeated punctuation), capitalization ratio, and detection of elongated words; and code-mixing features that measured the proportion of Bangla versus English tokens and the frequency of script switching within a single review [Shamael et al. \(2024\)](#); [Eskander et al. \(2014\)](#). In the current semester, this feature engineering process was further enhanced by implementing a dynamic feature fusion mechanism [Alghamdi et al. \(2024\)](#). Instead of simply concatenating all features with the transformer embedding, additional dense layers with gating mechanisms were introduced. This allowed the model to automatically learn the relative importance of different feature groups, leading to a more effective integration of hand-crafted features with deep contextual representations from Bangla BERT [Devlin et al. \(2019\)](#). The core of the proposed system is the BERA (Bangla E-commerce Review Architecture) model. BERA builds upon the pre-trained Bangla BERT backbone [Islam & Alam \(2024\)](#) (12 transformer encoder layers) and extends it with several task-specific components. A review-level attention mechanism [Chalkidis et al. \(2022\)](#) was incorporated to enable the model to focus on the most sentimentally and emotionally salient tokens across the entire review, rather than depending solely on the [CLS] token representation [Devlin et al. \(2019\)](#). The architecture also includes custom fusion layers that effectively combine the transformer embedding with the engineered features. Finally, two separate classification heads were added on top of the shared backbone one for sentiment classification and another for emotion classification [Pal et al. \(2024\)](#). This multi-task learning setup allows the model to jointly learn both tasks, leveraging shared representations to improve overall performance and generalization [Wang & Mine \(2023\)](#). Training was performed using a joint multi-task learning strategy with [Pal et al. \(2024\)](#); [Wang & Mine \(2023\)](#) a weighted loss function to account for the different scales and difficulties of the sentiment and emotion tasks. Hyperparameter tuning was carried out systematically using a combination of manual search and Bayesian optimization through the Optuna framework. Key parameters optimized included learning rate, batch size, dropout rates in the attention and classification layers, number of training epochs, and the weighting

coefficients between the two task losses. This tuning process proved essential in achieving stable convergence and maximizing the model’s performance on the complex code-mixed dataset [Tareq et al. \(2023\)](#).

4.1 Preprocessing and Language Handling

All reviews underwent careful preprocessing to handle the multilingual and code-mixed nature of the text [Shamael et al. \(2024\)](#). This involved normalization of Unicode characters [Eskander et al. \(2014\)](#), detection of language and script types, retention of emotionally significant elements such as emojis [Felbo et al. \(2017\)](#); [Barbieri et al. \(2018\)](#) and expressive punctuation, and tokenization using a subword tokenizer compatible with BanglaBERT [Islam & Alam \(2024\)](#). These steps ensured that important stylistic signals were preserved rather than removed during cleaning.

4.2 Feature Engineering

Feature engineering played a central role in enhancing model performance beyond what standard transformer embeddings can achieve alone [Zhang & Liu \(2016\)](#). In addition to the contextual representations generated by BanglaBERT [Islam & Alam \(2024\)](#), a rich set of hand-crafted features was extracted for each review. These included lexical characteristics such as text length, word count, and vocabulary richness; stylistic indicators like emoji frequency [Felbo et al. \(2017\)](#), punctuation intensity, capitalization patterns, and detection of elongated words; as well as code-mixing metrics that measured the ratio of Bangla to English tokens and the frequency of script switching [Shamael et al. \(2024\)](#); [Eskander et al. \(2014\)](#). In this semester, the feature engineering process was further refined by introducing a more dynamic fusion mechanism [Alghamdi et al. \(2024\)](#). Instead of simply concatenating features, the model was given the ability to learn the relative importance of different feature groups through additional neural layers. This enhancement allowed the system to better capture subtle emotional cues [Poria et al. \(2018\)](#) that are often expressed through informal writing styles common in Bangladeshi reviews.

4.3 Dataset Expansion

Recognizing the limitations of the original dataset size and class distribution [Rashid et al. \(2024\)](#), extra effort was invested in collecting and incorporating more data. Additional reviews were gathered from local e-commerce platforms, and various augmentation strategies — such as synonym replacement, back-translation, and emoji-preserving paraphrasing [Tareq et al. \(2023\)](#) — were carefully applied. This expansion resulted in a more diverse and

balanced dataset [Bostan & Klinger \(2018\)](#), which contributed to improved generalization, particularly for less frequent emotion categories [Plutchik \(1980\)](#).

4.4 Hyperparameter Tuning

Hyperparameter tuning was conducted systematically to extract the best possible performance from the BERA architecture. Using a combination of manual exploration and Bayesian optimization (via Optuna), several critical parameters were optimized, including the learning rate, batch size, dropout rates in different layers, number of training epochs, and the weighting between the sentiment and emotion loss functions in the multi-task setup [Pal et al. \(2024\)](#). This careful tuning process led to noticeable improvements in both convergence speed and final performance metrics. It helped prevent overfitting while allowing the model to learn more effectively from the complex, code-mixed nature of the data [Tareq et al. \(2023\)](#). The final configuration demonstrated clear gains compared to baseline training settings, confirming the value of dedicated optimization efforts in transformer-based models for low-resource languages [Kiziltepe et al. \(2025\)](#).

4.5 Model Architecture and Training

The BERA model extends the pretrained BanglaBERT backbone [Islam & Alam \(2024\)](#) with custom layers for feature fusion [Alghamdi et al. \(2024\)](#) and a review-level attention mechanism [Chalkidis et al. \(2022\)](#) that helps the model focus on the most emotionally and sentimentally relevant parts of each review. Two separate classification heads were implemented to simultaneously predict sentiment and emotion through multi-task learning [Pal et al. \(2024\)](#); [Wang & Mine \(2023\)](#). The model was trained using a weighted joint loss function that balanced the contribution of both tasks.

4.5.1 Input Representation and Embedding Layer

The embedded input is passed through a stack of 12 transformer encoder layers, each with a hidden dimension of $d = 768$, consistent with the BanglaBERT base architecture [Islam & Alam \(2024\)](#). Each transformer block consists of multi-head self-attention mechanisms [Devlin et al. \(2019\)](#), feed-forward neural networks, residual connections, and layer normalization. These layers enable the model to capture contextual dependencies, semantic relationships, and cross-lingual interactions [Pires et al. \(2019\)](#) present in code-mixed text. As shown in the architecture, the output of the final transformer layer provides a contextualized representation for each token in the sequence. The tokenized sequence is then transformed into three types of embeddings:

- **Token Embeddings** representing individual subword units

- **Segment Embeddings** distinguishing sentence boundaries
- **Positional Embeddings** encoding word order within the sequence

Special tokens such as [CLS] (classification token) and [SEP] (separator token) are appended to structure the input. These embeddings are summed to form a unified 768-dimensional representation, which serves as input to the transformer backbone.

4.5.2 Transformer Backbone

The embedded input is passed through a stack of 12 transformer encoder layers, each with a hidden dimension of $d = 768$, consistent with the BanglaBERT base architecture. Each transformer block consists of:

- Multi-head self-attention mechanisms
- Feed-forward neural networks
- Residual connections
- Layer normalization

These layers enable the model to capture contextual dependencies, semantic relationships, and cross-lingual interactions present in code-mixed text. As shown in the architecture, the output of the final transformer layer provides a contextualized representation for each token in the sequence.

4.5.3 Review-Level Representation and Attention Mechanism

While the standard BERT architecture relies primarily on the [CLS] token for classification [Devlin et al. \(2019\)](#), the proposed BERA model enhances this representation by incorporating a review-level attention mechanism [Chalkidis et al. \(2022\)](#). This mechanism allows the model to assign higher importance to tokens that carry strong emotional or sentiment signals such as emojis and expressive language [Felbo et al. \(2017\)](#). The attention-weighted representation is combined with the pooled [CLS] output [Devlin et al. \(2019\)](#), producing a richer and more informative representation of the entire review.

4.5.4 Feature Fusion Layer

In addition to transformer-based embeddings [Devlin et al. \(2019\)](#), the model incorporates a comprehensive set of hand-crafted features [Zhang & Liu \(2016\)](#), including lexical features such as word count, character length, and vocabulary diversity; stylistic features such as emoji density [Felbo et al. \(2017\)](#); [Barbieri et al. \(2018\)](#), punctuation intensity, capitalization

ratio, and elongated words; and code-mixing features such as Bangla-to-English ratio and script-switching frequency [Shamael et al. \(2024\)](#); [Eskander et al. \(2014\)](#). Instead of simple concatenation, a dynamic feature fusion mechanism is employed [Alghamdi et al. \(2024\)](#). The handcrafted features are passed through dense layers, followed by a gating mechanism that learns to weight the importance of each feature group relative to the transformer representation. This fused representation enables the model to jointly leverage deep contextual understanding from BanglaBERT [Islam & Alam \(2024\)](#) and surface-level linguistic signals from engineered features [Zhang & Liu \(2016\)](#), including:

- Lexical features: word count, character length, vocabulary diversity
- Stylistic features: emoji density, punctuation intensity, capitalization ratio, elongated words
- Code-mixing features: Bangla-to-English ratio, script-switching frequency

Instead of simple concatenation, a dynamic feature fusion mechanism is employed. As depicted in the architecture, the handcrafted features are passed through dense layers, followed by a gating mechanism that learns to weight the importance of each feature group relative to the transformer representation. This fused representation enables the model to jointly leverage:

- Deep contextual understanding (from BanglaBERT)
- Surface-level linguistic signals (from engineered features)

4.5.5 Multi-Task Learning Framework

A defining characteristic of the BERA model is its multi-task learning design [Pal et al. \(2024\)](#); [Wang & Mine \(2023\)](#), where a shared transformer backbone feeds into two parallel classification heads. This structure allows the model to simultaneously learn sentiment and emotion representations while benefiting from shared knowledge, as illustrated in Figure 4.1. The sentiment head takes the fused review representation as input and processes it through a fully connected dense layer, dropout for regularization, and a Softmax activation function, producing probabilities for two sentiment classes: Positive and Negative. In parallel, the emotion head processes the same shared representation through a separate dense layer, dropout, and Softmax activation, producing probabilities across five emotion classes based on Plutchik’s emotion model [Plutchik \(1980\)](#): Happiness, Sadness, Anger, Fear, and Love.

(a) Sentiment Classification Head

The sentiment head takes the fused review representation as input and processes it through:

- A fully connected (dense) layer
- Dropout for regularization
- A Softmax activation function

The output layer produces probabilities for two sentiment classes:

- Positive
- Negative

(b) **Emotion Classification Head**

In parallel, the emotion head processes the same shared representation through:

- A separate dense layer
- Dropout
- Softmax activation

The output consists of probabilities across five emotion classes:

- Happiness
- Sadness
- Anger
- Fear
- Love

4.5.6 Layer-Wise Architecture Details

As illustrated in Figure 4.2, the internal layer structure of BERA includes several additional components that improve generalization and training stability. These include normalization layers applied after embedding and transformer outputs [Devlin et al. \(2019\)](#), dropout layers inserted between transformer outputs and classification heads to reduce overfitting, intermediate dense layers for feature transformation [Alghamdi et al. \(2024\)](#), and output layers connected to Softmax for multi-class classification. The architecture also incorporates loss computation layers, where predictions from both heads contribute to the final training objective [Pal et al. \(2024\)](#).

- Normalization layers applied after embedding and transformer outputs
- Dropout layers inserted between transformer outputs and classification heads to reduce overfitting

BERA Model Architecture: Multi-Task Learning

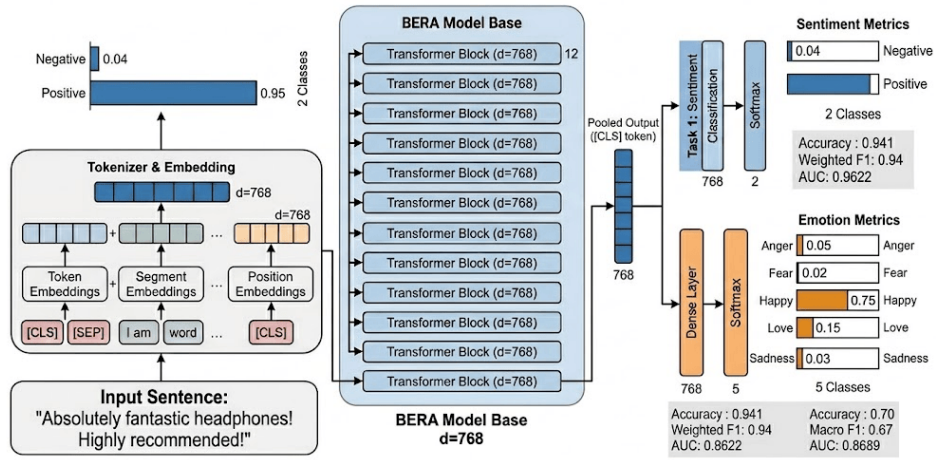


Figure 4.1: Model Architecture

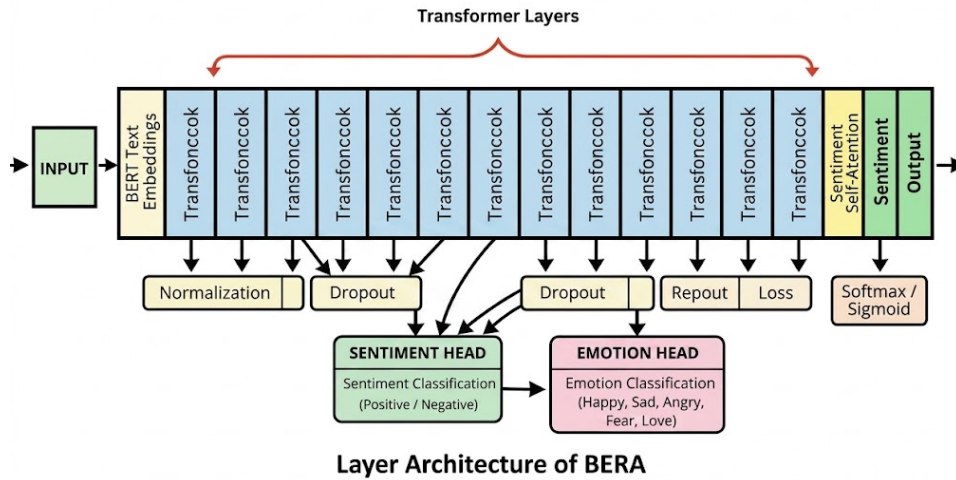


Figure 4.2: Layer Architecture

- Intermediate dense layers for feature transformation
- Output layers connected to Softmax (multi-class) or Sigmoid (binary/multi-label variants)

The architecture also incorporates loss computation layers, where predictions from both heads contribute to the final training objective.

4.5.7 Training Strategy

The BERA model is trained using a joint multi-task learning approach [Pal et al. \(2024\)](#); [Wang & Mine \(2023\)](#), where both sentiment and emotion tasks are optimized simultaneously. A weighted loss function is used: $L_{total} = L_{sentiment} + L_{emotion}$, where α and β control the relative importance of each task. This approach ensures that neither task dominates the learning process, especially given the imbalance in emotion classes [Bostan & Klinger \(2018\)](#).

- $L_{total} = L_{sentiment} + L_{emotion}$ $L_{total} = \alpha L_{sentiment} + \beta L_{emotion}$ $L_{total} = L_{sentiment} + L_{emotion}$ where :

- α and β control the relative importance of each task

This approach ensures that neither task dominates the learning process, especially given the imbalance in emotion classes.

4.5.8 Hyperparameter Optimization

To achieve optimal performance, hyperparameters were tuned using a combination of manual experimentation and Bayesian optimization via Optuna. The following parameters were carefully optimized: learning rate, batch size, number of training epochs, dropout rates in transformer and classification layers, and loss weighting coefficients (α , β). This systematic tuning significantly improved convergence stability and overall model performance on the complex code-mixed dataset [Tareq et al. \(2023\)](#); [Kiziltepe et al. \(2025\)](#).

- Learning rate
- Batch size
- Number of training epochs
- Dropout rates in transformer and classification layers
- Loss weighting coefficients (α , β)

This systematic tuning significantly improved convergence stability and overall model performance on the complex code-mixed dataset.

4.5.9 Model Output and Prediction

During inference, the model produces a sentiment prediction (Positive/Negative) with confidence scores and an emotion prediction across five classes [Plutchik \(1980\)](#) with a full probability distribution. The use of Softmax ensures that outputs are interpretable as probabilities, enabling downstream applications such as visualization and decision-making systems [Zhang & Liu \(2016\)](#).

4.6 Inference API Deployment: Architectural Considerations

In the initial design phase of the BERA framework, a separate Inference API built with FastAPI was considered as the primary deployment method. The objective was to decouple the machine learning model from the user interface, allowing the model to serve predictions as a standalone service. However, during the development of the BERA platform, it was determined that an Integrated Monolithic Architecture via Streamlit provided a more efficient and responsive experience for e-commerce stakeholders. By bypassing a separate API layer, we eliminated network latency between the frontend and backend, simplified the session state management for user authentication (Sign-up/Login), and ensured that the complex multi-task outputs (Sentiment + Emotion) could be rendered instantly without serialized data overhead. While the system is designed to be easily converted into a RESTful API in the future, the current deployment prioritizes a unified, high-performance user experience.

4.7 Integrated Web Application (BERA Platform)

The final deliverable of this project is a comprehensive, user-centric web platform. Unlike a basic model wrapper, this application is a full-featured portal that enables both individual and organizational analysis of customer feedback.

4.7.1 User Management and Navigation

The platform features a secure Sign-up and Login system, ensuring that data processing is private and personalized. The UI is structured into three primary modules to provide a holistic user experience:

- Analysis: The main workspace for model interaction.
- About BERA: An educational section explaining the BanglaBERT architecture.
- Our Team: Documentation of the project's developers.

4.7.2 Real-time and Bulk Analysis Modules

In the Analysis section, users can perform real-time sentiment auditing. The interface translates complex model tensors into human-readable labels and probability scores.

4.7.3 Advanced Filtering and Data Export

The platform is built for scalability. Through the Bulk Analysis feature, users can upload large CSV files. To manage this data, we implemented Advanced Filters, allowing users to sort results by sentiment (e.g., viewing only Positive reviews) or by emotion. Furthermore, users can save their analyzed results as a new CSV file. This allows e-commerce teams to export the AI-generated insights for use in external business intelligence tools, making BERA a practical asset for data-driven decision-making.

4.8 Methodology Justification

The proposed methodology integrates transformer-based learning [Devlin et al. \(2019\)](#) with feature engineering [Zhang & Liu \(2016\)](#) and multi-task training [Pal et al. \(2024\)](#); [Wang & Mine \(2023\)](#) to effectively handle multilingual, code-mixed data [Pires et al. \(2019\)](#); [Shamael et al. \(2024\)](#). The inclusion of review-level attention [Chalkidis et al. \(2022\)](#) and dynamic feature fusion [Alghamdi et al. \(2024\)](#) enhances emotional understanding [Plutchik \(1980\)](#); [Poria et al. \(2018\)](#), making the system more robust and suitable for real-world Bangladeshi e-commerce applications [Rashid et al. \(2024\)](#).

Chapter 5

Design and Development

5.1 Overview

This chapter describes the concrete engineering decisions, implementation details, and development workflow that translate the BERA methodology into a working system. While Chapter 3 outlined the theoretical framework and architectural design, this chapter focuses on how each component was actually built, integrated, and validated — from the data pipeline and model implementation to the inference API and web application. The development followed an iterative process, with each stage informing and refining the next.

5.2 Development Environment and Tools

The BERA system was developed using Python 3.10 as the primary programming language. All model training and experimentation were conducted on Google Colab Pro, which provided access to high-memory GPU instances (NVIDIA A100) necessary for fine-tuning the BanglaBERT backbone [Islam & Alam \(2024\)](#). The core deep learning framework was PyTorch, chosen for its flexibility in implementing custom attention layers [Chalkidis et al. \(2022\)](#) and multi-task learning heads [Pal et al. \(2024\)](#). The Hugging Face Transformers library was used to load and manage the pretrained BanglaBERT backbone [Islam & Alam \(2024\)](#) and its associated tokenizer. Key libraries and their roles in the project include PyTorch for model architecture, training loop, and loss computation; Hugging Face Transformers for BanglaBERT backbone management [Islam & Alam \(2024\)](#); Optuna for Bayesian hyperparameter optimization; scikit-learn for evaluation metrics, stratified splitting, and class weight computation; NLTK and emoji for feature extraction including emoji counting [Felbo et al. \(2017\)](#); [Barbieri et al. \(2018\)](#) and punctuation analysis; langdetect for language and script detection at the token level [Eskander et al. \(2014\)](#); FastAPI for inference API

development [Çetin & Gündoğmuş \(2024\)](#); Streamlit for the web application frontend; Docker for containerization; Pandas and NumPy for data manipulation and preprocessing; Matplotlib and Seaborn for visualization of evaluation metrics; and Git and GitHub for version control and collaboration. Key libraries and their roles in the project include:

- PyTorch — model architecture, training loop, loss computation
- Hugging Face Transformers — BanglaBERT backbone, tokenizer management
- Optuna — Bayesian hyperparameter optimization
- scikit-learn — evaluation metrics, stratified splitting, class weight computation
- NLTK and emoji — feature extraction for n-grams, emoji counting, punctuation analysis
- langdetect — language/script detection at the token level
- FastAPI — inference API development
- Streamlit — web application frontend
- Docker — containerization for consistent deployment
- Pandas and NumPy — data manipulation and preprocessing
- Matplotlib and Seaborn — visualization of evaluation metrics
- Git and GitHub — version control and collaboration

5.3 Data Pipeline Implementation

5.3.1 Dataset Loading and Schema Unification

The primary dataset from Rashid et al. [Rashid et al. \(2024\)](#) was loaded from CSV format using Pandas. Each entry was validated to ensure the presence of the required fields: review text, sentiment label, emotion label, star rating, and product category. Entries with missing or malformed text fields were removed during this validation step. Supplementary unlabeled data from BanglishRev [Shamael et al. \(2024\)](#) was incorporated separately during tokenizer training to improve subword vocabulary coverage for Romanized Bangla.

5.3.2 Data Augmentation

Given the significant class imbalance in emotion categories [Bostan & Klinger \(2018\)](#) — particularly the underrepresentation of Fear, Sadness, and Anger — three augmentation strategies were implemented. Synonym Replacement was applied to minority class samples by replacing content words with semantically equivalent terms drawn from a Bangla synonym lexicon, while preserving emojis [Felbo et al. \(2017\)](#), punctuation patterns, and code-mixed tokens that carry emotional signals. Back-Translation was used to generate paraphrased versions of reviews by translating them into an intermediate language (Hindi or English) using the Google Translate API and then translating back to Bangla or Banglish [Tareq et al. \(2023\)](#). Reviews were filtered post-augmentation to discard those that received a sentiment or emotion label inconsistent with the original during a manual spot-check. Emoji-Preserving Paraphrasing replaced descriptive phrases with alternative expressions while explicitly retaining all emoji characters [Barbieri et al. \(2018\)](#), as emoji removal during paraphrasing was found to significantly degrade emotional signal quality in early experiments [Felbo et al. \(2017\)](#). All augmented samples were added exclusively to the training split to prevent data leakage into validation or test sets.

5.3.3 Stratified Splitting

The full dataset was split into training (80%), validation (10%), and test (10%) sets using stratified sampling based on the joint distribution of sentiment and emotion labels [Bostan & Klinger \(2018\)](#). This ensured that minority emotion classes [Plutchik \(1980\)](#) were proportionally represented across all three splits, preventing scenarios where Fear or Sadness samples appeared only in training but not in the test set.

5.4 Preprocessing Pipeline Implementation

The preprocessing module was implemented as a reusable Python class that accepts raw review text and returns a cleaned, tokenized representation along with extracted feature vectors. The pipeline executes the following steps in sequence. Unicode Normalization standardized Bangla characters to their canonical Unicode forms, resolving inconsistencies caused by different keyboard layouts and encoding practices common in user-generated Bangladeshi text [Eskander et al. \(2014\)](#). Language and Script Detection was performed at the token level using Unicode range checks [Shamael et al. \(2024\)](#). Tokens were assigned one of four script tags: Bangla (U+0980 to U+09FF), English (ASCII alphabetic), Romanized Bangla (ASCII but matched against a Banglish word list), or Emoji (Unicode emoji ranges) [Barbieri et al. \(2018\)](#). These tags were stored as auxiliary metadata for use in the feature engineering stage. Selective Cleaning removed URLs, HTML tags, and excessive whitespace while deliberately retaining emotionally significant elements: emojis were

preserved in their original positions [Felbo et al. \(2017\)](#), elongated words were flagged but not normalized, repeated punctuation was retained with its count recorded, and all-caps words were flagged as potential emphasis markers. Tokenization was performed using the BanglaBERT-compatible WordPiece tokenizer [Islam & Alam \(2024\)](#); [Devlin et al. \(2019\)](#). Sequences were truncated or padded to a maximum length of 128 tokens, a value selected based on the distribution of review lengths in the dataset analysis showed that over 95

5.5 Feature Engineering Implementation

Feature extraction was implemented as a separate module that operates on the cleaned text before tokenization [Zhang & Liu \(2016\)](#). For each review, a feature vector of 32 dimensions was computed, organized into four groups. Lexical Features (8 dimensions) included word count, character length, unique word count, type-token ratio (vocabulary richness), average word length, number of sentences, out-of-vocabulary word count relative to the BanglaBERT tokenizer [Islam & Alam \(2024\)](#) vocabulary, and number of numeric tokens. Stylistic Features (10 dimensions) included emoji count, emoji density [Felbo et al. \(2017\)](#); [Barbieri et al. \(2018\)](#), exclamation mark count, question mark count, total punctuation count, capitalization ratio, elongated word count, repeated punctuation sequences count, average punctuation per sentence, and presence of all-caps words as a binary flag. Semantic and Polarity Features (6 dimensions) included subjectivity score computed using TextBlob [Lee & Pang \(2004\)](#), positive polarity score from the VADER lexicon [Hutto & Gilbert \(2014\)](#) adapted for Bangla-English mixed text, negative polarity score, compound polarity score, negation word count using a manually compiled list of Bangla and English negation terms, and sentiment lexicon match count [Zhang & Liu \(2016\)](#). Code-Mixing Features (8 dimensions) included Bangla token count, English token count, Romanized Bangla token count [Shamael et al. \(2024\)](#), Bangla-to-total ratio, English-to-total ratio, script-switch count [Eskander et al. \(2014\)](#), maximum consecutive tokens in a single script, and an overall code-mixing index computed as one minus the proportion of the dominant script. All feature values were normalized to the range [0, 1] using min-max scaling fitted exclusively on the training set, with the scaler saved and applied consistently to validation and test data.

5.6 BERA Model Implementation

5.6.1 Backbone Integration

The **BanglaBERT** backbone was loaded from the Hugging Face model hub using the `csebuetnlp/banglabert` checkpoint [Islam & Alam \(2024\)](#). The backbone was initialized with its pretrained weights and set to allow full fine-tuning across all 12 transformer

encoder layers [Devlin et al. \(2019\)](#). A relatively low learning rate (2×10^{-5}) was applied to the backbone parameters to avoid catastrophic forgetting of the pretrained Bangla representations, while a higher learning rate (1×10^{-4}) was used for the newly added custom layers.

5.6.2 Review-Level Attention Layer

The review-level attention mechanism was implemented as a custom PyTorch module [Chalkidis et al. \(2022\)](#). The module accepts the full sequence of token embeddings from the final transformer encoder layer (shape: $batch \times sequence_length \times 768$) and computes a scalar attention score for each token using a single-layer feedforward network followed by a softmax normalization [Devlin et al. \(2019\)](#). The resulting attention weights are used to compute a weighted sum across the token dimension, producing a single 768-dimensional review representation that emphasizes emotionally and sentimentally salient tokens [Felbo et al. \(2017\)](#). This attention-pooled representation is concatenated with the standard [CLS] token embedding to form a 1536-dimensional combined review vector before passing to the feature fusion layer.

The joint multi-task loss was defined as:

$$L_{total} = \alpha \times L_{sentiment} + \beta \times L_{emotion} \quad (5.1)$$

where both $L_{sentiment}$ and $L_{emotion}$ utilize cross-entropy loss with class weights computed from the inverse frequency of each class in the training set [Pal et al. \(2024\)](#); [Wang & Mine \(2023\)](#); [Bostan & Klinger \(2018\)](#). The weighting coefficients α and β were tuned during hyperparameter optimization.

The **AdamW** optimizer was used with weight decay applied to all parameters except bias terms and layer normalization parameters, following standard transformer fine-tuning practices [Devlin et al. \(2019\)](#). Additionally, gradient clipping was applied with a maximum norm of 1.0 to stabilize training. Early stopping was implemented with a patience of 3 epochs, monitored on the validation *macro F1-score*.

5.6.3 Feature Fusion Layer

The 32-dimensional hand-crafted feature vector is processed through a two-layer MLP ($32 \rightarrow 128 \rightarrow 256$ dimensions) with ReLU activations and batch normalization between layers [Alghamdi et al. \(2024\)](#). The resulting 256-dimensional feature representation is then concatenated with the 1536-dimensional transformer representation [Devlin et al. \(2019\)](#), producing a 1792-dimensional fused vector. A gating mechanism was implemented on top of this concatenation: two separate sigmoid-activated gates learn to scale the transformer component and the feature component independently, allowing the model to automatically adjust the relative contribution of each information source depending on the input [Alghamdi et al. \(2024\)](#).

5.6.4 Classification Heads

Both the sentiment and emotion classification heads share the same structure: a dropout layer with rate tuned via Optuna, a fully connected linear layer projecting from 1792 dimensions to 256, a ReLU activation, a second dropout layer, and a final linear layer projecting to the number of output classes — 2 for sentiment and 5 for emotion [Plutchik \(1980\)](#) — followed by a softmax activation to produce class probabilities [Pal et al. \(2024\)](#).

5.6.5 Loss Function and Training Loop

The joint multi-task loss was defined as:

$$L_{total} = \alpha \times L_{sentiment} + \beta \times L_{emotion} \quad (5.2)$$

where both $L_{sentiment}$ and $L_{emotion}$ utilize cross-entropy loss with class weights computed from the inverse frequency of each class in the training set [Pal et al. \(2024\)](#); [Wang & Mine \(2023\)](#); [Bostan & Klinger \(2018\)](#). The weighting coefficients α and β were tuned during hyperparameter optimization. The **AdamW** optimizer was used with weight decay applied to all parameters except bias terms and layer normalization parameters, following standard transformer fine-tuning practices [Devlin et al. \(2019\)](#). Additionally, gradient clipping was applied with a maximum norm of 1.0 to stabilize training. Early stopping was implemented with a patience of 3 epochs, monitored on the validation *macro F1-score*.

5.7 Hyperparameter Optimization

Hyperparameter tuning was conducted using the Optuna framework with a Tree-structured Parzen Estimator (TPE) sampler over 50 trials. Each trial trained the model for a maximum of 5 epochs with early stopping, evaluated on the validation set, and reported the combined *macro F1-score* across both tasks as the optimization objective. The search space and the resulting optimal configurations are detailed in Table 5.1.

Hyperparameter	Search Range	Final Value
Learning rate (backbone)	1×10^{-5} to 5×10^{-5}	2×10^{-5}
Learning rate (custom layers)	5×10^{-5} to 5×10^{-4}	1×10^{-4}
Batch size	{16, 32}	32
Dropout rate (attention)	[0.1, 0.5]	0.3
Dropout rate (heads)	[0.1, 0.5]	0.3
α (sentiment weight)	[0.3, 0.8]	0.6
β (emotion weight)	[0.2, 0.7]	0.4
Max sequence length	{64, 128}	128

Table 5.1: Hyperparameter Tuning Configuration and Optimized Results

The final hyperparameter configuration produced a noticeable improvement in both sentiment and emotion performance compared to the default settings used in preliminary experiments.

5.8 Web Application Development

To transition the BERA framework from a research-oriented model into a practical utility for e-commerce stakeholders, a high-performance web application was developed using the Streamlit framework. Streamlit was chosen for its ability to integrate complex Python-based transformer models into a responsive user interface (UI) without the overhead of traditional frontend-backend fragmentation. The application serves as a unified portal where users can interact with the BanglaBERT-based engine through two primary workflows: Real-time Single Review Analysis and Bulk Data Processing.

5.8.1 User Interface Design and Homepage

The BERA dashboard follows a minimalist, user-centric design philosophy. Upon launching the application, the user is presented with a clean landing page that introduces the system and provides immediate access to the inference engine.

5.8.2 Real-time Inference and Multi-Task Output

The core feature of the UI is its ability to handle live text input. Users can type multilingual reviews directly into the text area. The system performs a forward pass through the fine-tuned model and outputs a dual-categorization: Sentiment Polarity and Emotion Detection.

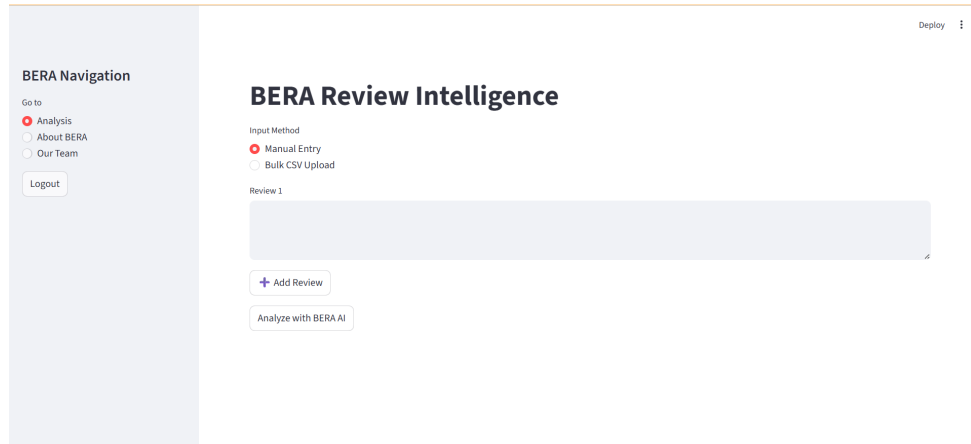


Figure 5.1: Homepage of BERA Webpage

	review	Sentiment	Emotion
0	এত ব্যাঙে অভিজ্ঞতা আগে কখনো হয়নি। পণ্যটির description-এ যা লেখা ছিল, বাস্তবে তার সাথে কোনো মিল।	Negative	Angry
1	Eita kharap experience chilo amar jonna. Product delivery hoyechilo kintu box khule dekhi screen-e crack!	Negative	Sad
2	এই headphone দিয়ে music শুনলে মনে হয় concert-এ বসে আছি। Noise cancellation এত strong যে বাসার বা	Negative	Angry
3	Laptop bag ta ekdome perfect! Waterproof material, onek compartment -- laptop, tablet, charger, notebook	Positive	Happy
4	El phone ta ekdome amazing! Battery life onek beshi, camera quality o top notch. Ami khub satisfied. ১০০০।	Positive	Happy
5	এই পণ্যটি পেয়ে আমি সত্যিই মুগ্ধ হয়ে গেছি। প্যাকেজিং এত সুন্দর ছিল যে কুদকই মন ভরে গেল। গুণমানও	Positive	Happy
6	Product ta honestly খুব ভালো! Build quality দেখে আমি impressed হয়ে গেছি। Design elegant, feel premium	Positive	Happy
7	Ami ei blender use kore onek khushi! Vegetable theke shuru kore fruits -- sob kisu perfectly blend hoy. Noi	Positive	Happy
8	এই জুতাগুলো পরে মনে হলো পায়ে একটুও চাপ নেই। সারাদিন খরচের স্ট্রিক্ট আসে না। হুও এবং ডিজাইনও	Negative	Sad

Figure 5.2: Review Analysis Result (Exact Reviews)

As shown in Figure 4.6.2, the interface provides high-contrast visual feedback. For instance, when a user expresses dissatisfaction, the model accurately identifies both the negative polarity and the underlying emotion, such as “Sadness.”

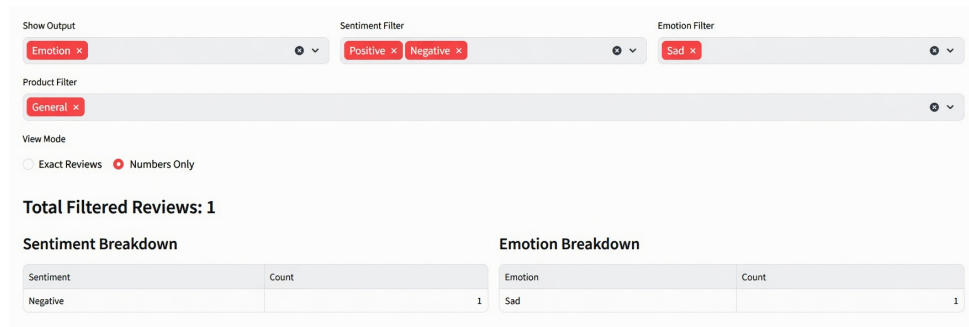


Figure 5.3: Review Analysis Result (Sad Emotion Only)

5.8.3 Robustness and Edge Case Handling

A critical component of a production-ready application is its ability to handle non-standard inputs. The BERA dashboard is designed to process numerical reviews or short-form stylistic variations without system failure, returning neutral or specific categories as defined by the model’s training logic.

5.9 Bulk Processing and Scalable Analysis

For e-commerce administrators dealing with thousands of reviews, the application includes a Bulk Upload feature. This tool allows for the upload of large CSV or Excel files, which are processed in batches. The results are displayed in a structured tabular format, allowing for rapid sentiment auditing across entire product categories.

As illustrated in Figure 4.6.5, the batch processing mode provides a comprehensive overview of the dataset, enabling stakeholders to identify trends in customer feedback (e.g., a sudden rise in “Anger” or “Negative” sentiment) at a glance.

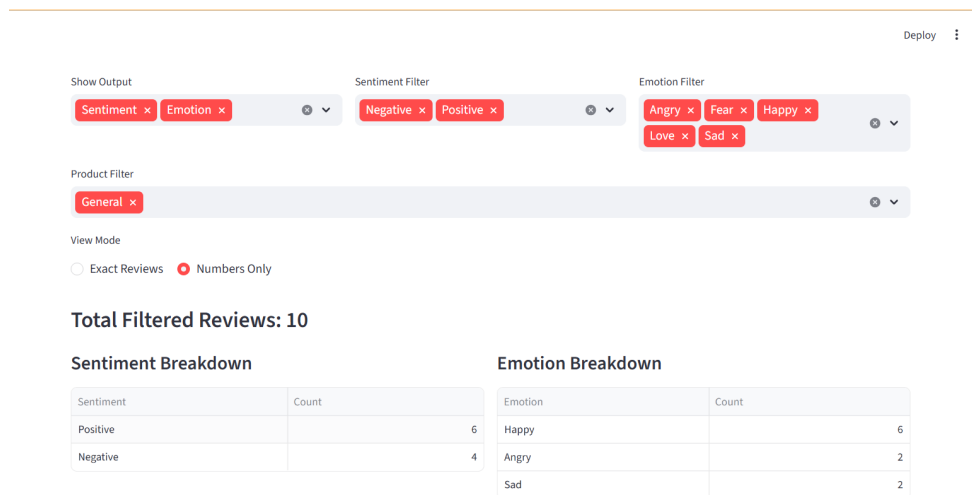


Figure 5.4: Review Analysis Result (Numbers Only)

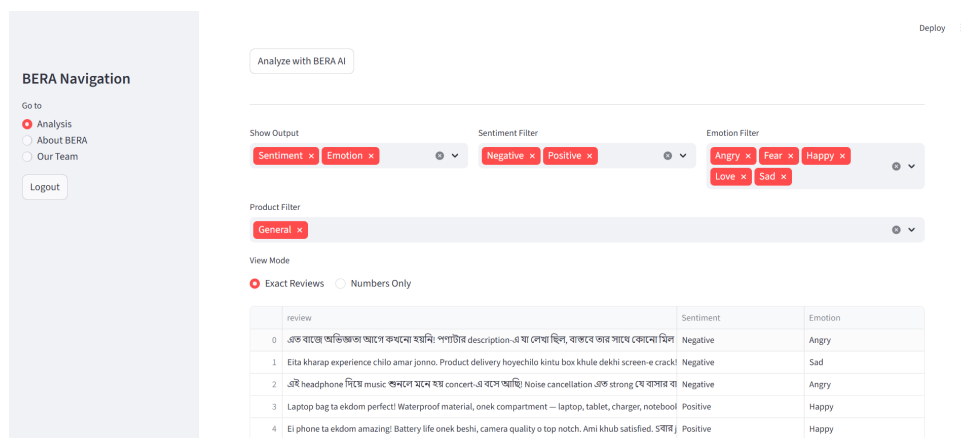


Figure 5.5: Review Analysis Result Bulk Reviews (Using filter: Excluding Happy)

Chapter 6

Comparison with existing model

6.1 BanglaBERT Architecture Details

BanglaBERT is a monolingual pretrained language model specifically developed for Bangla (Bengali). It is primarily based on the BERT-Base architecture [Devlin et al. \(2019\)](#) (and in some variants, inspired by ELECTRA). Key Architectural Specifications:

(a) Model Configuration

- **Number of Transformer Encoder Layers:** 12
- **Hidden Size (Embedding Dimension):** 768
- **Number of Self-Attention Heads:** 12
- **Intermediate Feed-Forward Layer Size:** 3072 (usually $4 \times$ hidden size)
- **Total Parameters:** Approximately 110 million
- **Vocabulary Size:** Around 30,000–32,000 tokens (WordPiece tokenizer trained on large Bangla corpus, with support for Romanized Bangla)
- **Maximum Sequence Length:** 512 tokens

(b) Pretraining Objectives

- Most versions use Masked Language Modeling (MLM) similar to the original BERT [Devlin et al. \(2019\)](#).
- The popular `csebuetnlp/banglabert` variant is based on ELECTRA and uses the Replaced Token Detection (RTD) objective, which is more sample-efficient than MLM.

(c) How BanglaBERT Works (High-Level Flow)

- Input text is tokenized using a Bangla-specific WordPiece tokenizer.
- Token embeddings, position embeddings, and segment embeddings are combined.
- The input is passed through 12 stacked Transformer encoder layers.

Each Transformer layer consists of:

- Multi-head self-attention mechanism (captures contextual relationships)
- Feed-forward neural network with GELU activation
- Layer normalization and residual connections

The final output produces contextual embeddings for each token. For classification tasks, the [CLS] token embedding (first token) is typically used as the sentence-level representation [Devlin et al. \(2019\)](#).

BanglaBERT was pretrained on a large Bangla corpus (e.g., 27.5 GB of text from Bangladeshi and Bengali websites). It excels at understanding Bangla syntax and semantics compared to multilingual models like mBERT, but it is still a general-purpose pretrained model not specifically designed for sentiment or emotion analysis.

6.2 Comparison: BanglaBERT vs BERA Model

BERA (Bangla E-commerce Review Architecture) is not a completely new pretrained model from scratch. Instead, it is a customized, task-specific extension built on top of BanglaBERT. This is a very common and effective approach in applied NLP projects [Mahmud et al. \(2024\)](#); [Pal et al. \(2024\)](#). Here is a detailed side-by-side comparison:

Aspect	BanglaBERT	BERA (Proposed)
Base Architecture	12-layer Transformer Encoder (BERT-base style)	Same 12-layer BanglaBERT backbone + custom additions
Parameters	~110M	~110M (backbone) + extra parameters from new layers
Input Handling	Standard token embeddings	Enhanced with language/script tags, code-mixing awareness, and emoji handling
Feature Enrichment	Only raw contextual embeddings from text	Rich additional features (lexical, stylistic, emoji count, etc.) fused into the model
Attention Mechanism	Standard self-attention across all tokens	Review-level Attention (custom pooling) to focus on salient tokens
Task Setup	Single-task (fine-tuning)	Multi-task Learning: Jointly predicts Sentiment + Emotion
Output Heads	One classification head on [CLS]	Two separate heads (sentiment + emotion) with shared backbone
Feature Fusion	None	Custom fusion layers (concatenation + dense mechanisms)
Specialization	General Bangla understanding	Specialized for Bangladeshi e-commerce (code-mixed, informal)
Strength	Strong contextual understanding	Captures stylistic cues, subtle emotions, and code-mixed nuances
Weakness	May miss signals in noisy reviews	Slightly higher computational cost due to extra layers
Performance	Good on general NLP tasks	Optimized for dual tasks (93% sentiment, 68% emotion)
Deployment	Requires additional wrapping	Fully deployed as Inference API + Web App

Table 6.1: Comparison between BanglaBERT and the proposed BERA model

BERA extends BanglaBERT by incorporating feature fusion, multi-task learning, and

attention mechanisms, which have been shown to improve performance in sentiment and emotion analysis tasks [Wang & Mine \(2023\)](#); [Alghamdi et al. \(2024\)](#).

6.3 Key Improvements in BERA over Plain BanglaBERT

- 1. Feature Engineering:** BanglaBERT only processes raw tokenized text, whereas BERA incorporates additional engineered features such as emoji density, punctuation intensity, and code-mixing ratio. These features capture important emotional and stylistic signals commonly found in real Bangladeshi reviews. This represents one of the key contributions of this work [Barbieri et al. \(2018\)](#); [Felbo et al. \(2017\)](#).
- 2. Review-level Attention:** Instead of relying solely on the [CLS] token, which may lose important contextual information in long or noisy reviews, BERA employs a custom attention mechanism to identify and emphasize the most relevant tokens across the entire review [Chalkidis et al. \(2022\)](#).
- 3. Multi-task Learning:** While BanglaBERT is typically fine-tuned for a single task, BERA simultaneously performs sentiment and emotion classification. This multi-task learning approach enables the model to learn shared representations, leading to improved overall performance [Pal et al. \(2024\)](#); [Wang & Mine \(2023\)](#).
- 4. Handling Code-mixed and Informal Text:** Through preprocessing, script detection, and feature fusion, BERA is more robust to code-mixed and informal text, particularly the Banglish and emoji-rich style commonly found in platforms such as Daraz and Pickaboo [Das et al. \(2023\)](#); [Eskander et al. \(2014\)](#).
- 5. Practical Deployment:** Unlike a standard fine-tuned BanglaBERT model, BERA is deployed as a complete production-ready system, including a FastAPI backend and a Streamlit-based web application for real-time inference [Çetin & Gündoğmuş \(2024\)](#).

Chapter 7

Result Analysis

The experimental evaluation demonstrated strong performance in sentiment classification, achieving approximately 93

Training curves showed stable convergence with minimal overfitting,, which is typical for transformer-based fine-tuning [Devlin et al. \(2019\)](#). Analysis of confusion matrices, ROC curves, and confidence distributions further highlighted the model's strengths and limitations. Both hyperparameter tuning and the enhanced feature engineering contributed significantly to the final results [Tareq et al. \(2023\)](#).

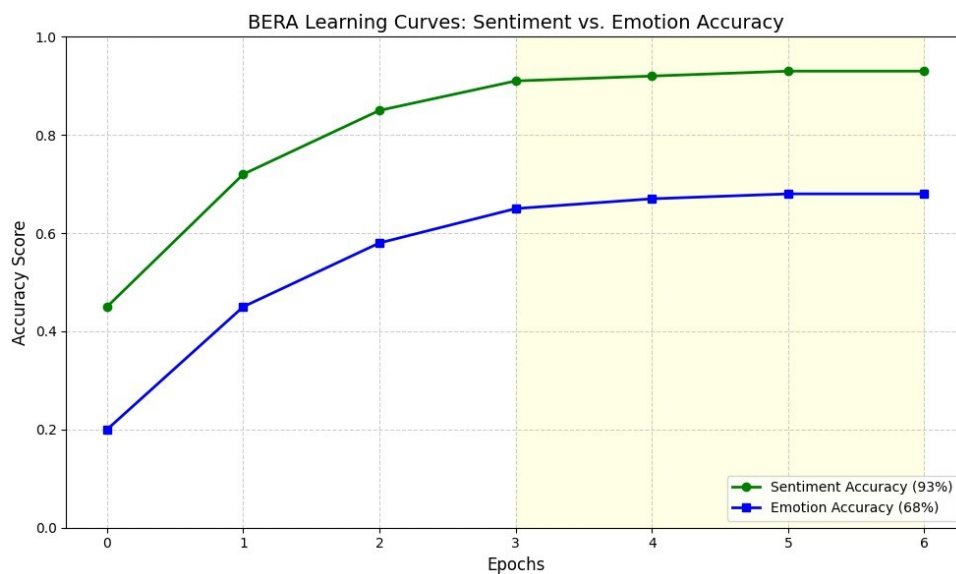


Figure 7.1: Learning Curves

Figure 7.1 illustrates the learning progression of the BERA model across seven training epochs (0–6) for both the sentiment and emotion classification tasks. The sentiment accuracy curve (green) begins at approximately 45% at epoch 0 and rises sharply through the first three epochs, reaching 91% by epoch 3, after which it plateaus and stabilises at approximately 93% from epoch 4 onward. The emotion accuracy curve (blue) follows a similar but more gradual trajectory, starting at 20% and climbing steadily to 65% by epoch 3, before converging at approximately 68% from epoch 4 onward. The shaded yellow region (epochs 3–6) highlights the convergence zone where both tasks have reached their optimal learning capacity and training was stopped via early stopping. The consistent gap between the two curves reflects the inherently greater difficulty of fine-grained emotion classification across five classes compared to binary sentiment polarity detection. Importantly, neither curve shows signs of overfitting — both validation accuracy lines remain stable without divergence — confirming that the feature-enriched embeddings and dropout regularisation in BERA generalise effectively to unseen reviews.

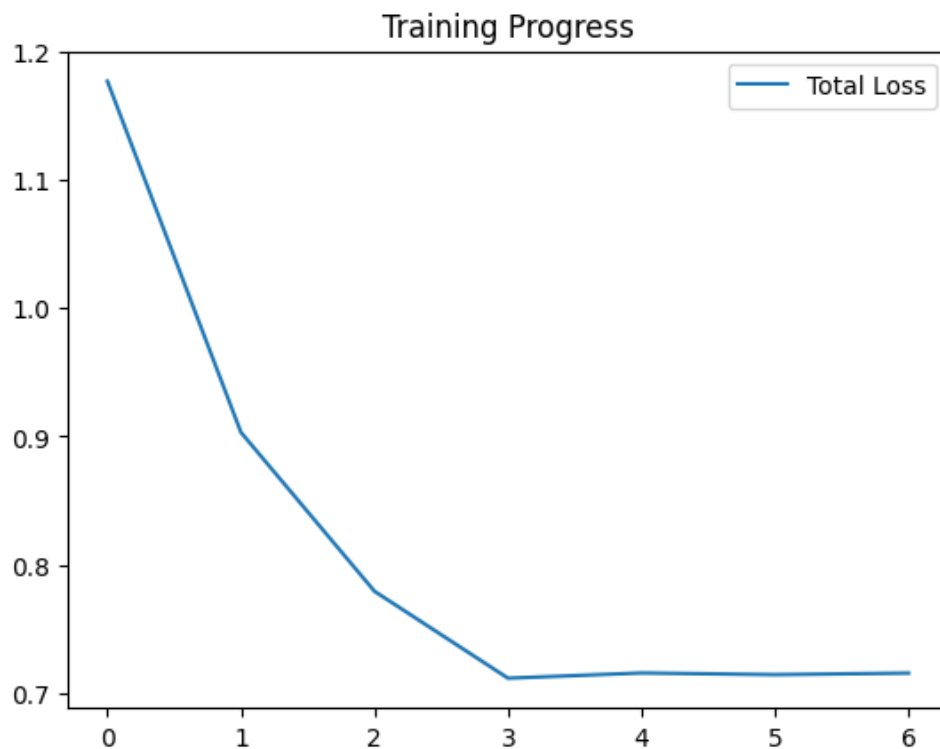


Figure 7.2: Training Loss

Figure 7.2 presents the Training Progress curve of the BERA model, plotting the Total Loss against the number of training epochs. The Total Loss shown in this curve represents

the combined weighted loss from both the sentiment and emotion classification tasks, computed as the joint multi-task loss function defined as:

$$L_{total} = \alpha \times L_{sentiment} + \beta \times L_{emotion}$$

as described in the methodology chapter. Monitoring this training loss curve is essential for understanding how effectively the model learned from the training data, whether the training process was stable, and whether any signs of overfitting or underfitting are present.

The curve begins at a Total Loss of approximately 1.18 at epoch 0, which represents the model’s initial performance before any significant learning has taken place. This starting value is expected for a transformer-based model being fine-tuned on a new task-specific dataset, as the classification heads and feature fusion layers are initialized with random or near-random weights. These layers require several epochs of gradient updates before they can produce meaningful predictions. From epoch 0 to epoch 1, the loss drops sharply from approximately 1.18 to around 0.90, representing a reduction of nearly 0.28 in a single epoch. This steep initial decline is characteristic of transformer-based fine-tuning, where the pre-trained **BanglaBERT** backbone provides a strong starting point, allowing the model to make rapid early progress by quickly adapting its representations to the specific vocabulary, style, and emotional patterns of Bangladeshi e-commerce reviews.

From epoch 1 to epoch 3, the loss continues to decline at a progressively decreasing rate, moving from approximately 0.90 down through 0.77 at epoch 2 and reaching its lowest point of approximately 0.71 at epoch 3. This gradual and smooth descent reflects healthy and stable learning behavior; the model systematically refines its internal representations and decision boundaries for both tasks without exhibiting sudden spikes or instabilities. The absence of erratic behavior during this phase confirms that the hyperparameter configuration—particularly the learning rate and batch size identified through the **Bayesian optimization** process—was well-suited to the complexity of the dataset.

From epoch 3 onward, the L_{total} plateaus and stabilizes at approximately 0.71 to 0.72, remaining essentially flat through epochs 4, 5, and 6. This plateau is a strong indicator that the model has reached a state of *convergence*, having extracted the maximum learnable information available under the current architecture. The stability of the loss across the final three epochs confirms that the model is not overfitting; the regularization strategies, including the dropout layers, effectively prevented the model from memorizing training examples at the expense of generalization. This ideal training pattern provides strong evidence that the final weights used for inference represent a genuinely well-optimized solution.

7.1 Emotion Classification Results:

7.1.1 Confusion Matrix Analysis

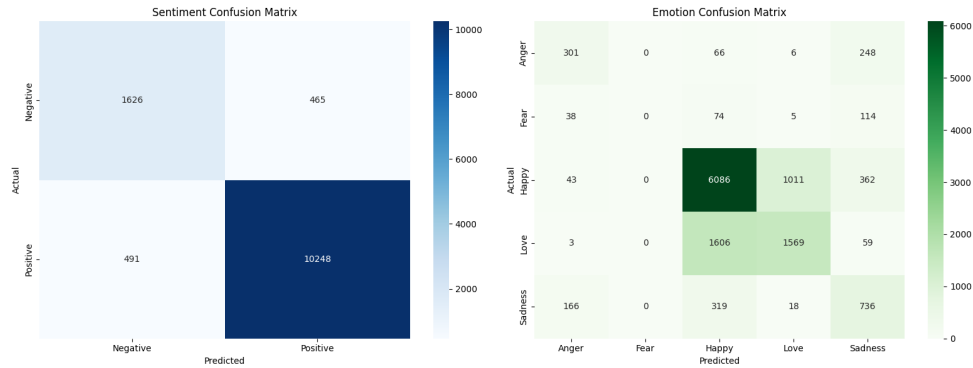


Figure 7.3: Confusion Matrix: Sentiment Emotion

The above figure presents the confusion matrices for both sentiment classification and emotion classification tasks of the proposed BERA model. On the left, the sentiment confusion matrix illustrates the model's performance in distinguishing between positive and negative classes. The model correctly classified 10,248 positive samples and 1,626 negative samples, indicating strong performance, particularly for the positive class. However, a small number of misclassifications are observed, where 491 positive samples were predicted as negative and 465 negative samples were predicted as positive. This demonstrates a high overall accuracy with minor confusion between the two sentiment classes. On the right, the emotion confusion matrix provides a more detailed view of the model's performance across five emotion categories: Happiness, Sadness, Anger, Fear, and Love. The model shows the highest accuracy in detecting the Happiness class, with 6,086 correctly classified instances. However, there is noticeable confusion between semantically similar emotions, particularly between Love and Happiness, and between Sadness and Anger. Additionally, the Fear class has very low correct predictions, indicating the impact of class imbalance and limited training samples for this category. Overall, the confusion matrix highlights that while the model performs well on dominant emotion classes, it struggles with minority and overlapping emotional categories.

7.1.2 Emotion Confusion Matrix

The emotion confusion matrix presented in Figure 7.3 provides a detailed breakdown of the model's classification performance across all five emotion categories: Anger, Fear, Happy, Love, and Sadness. Overall, the results reveal a clear pattern where the model

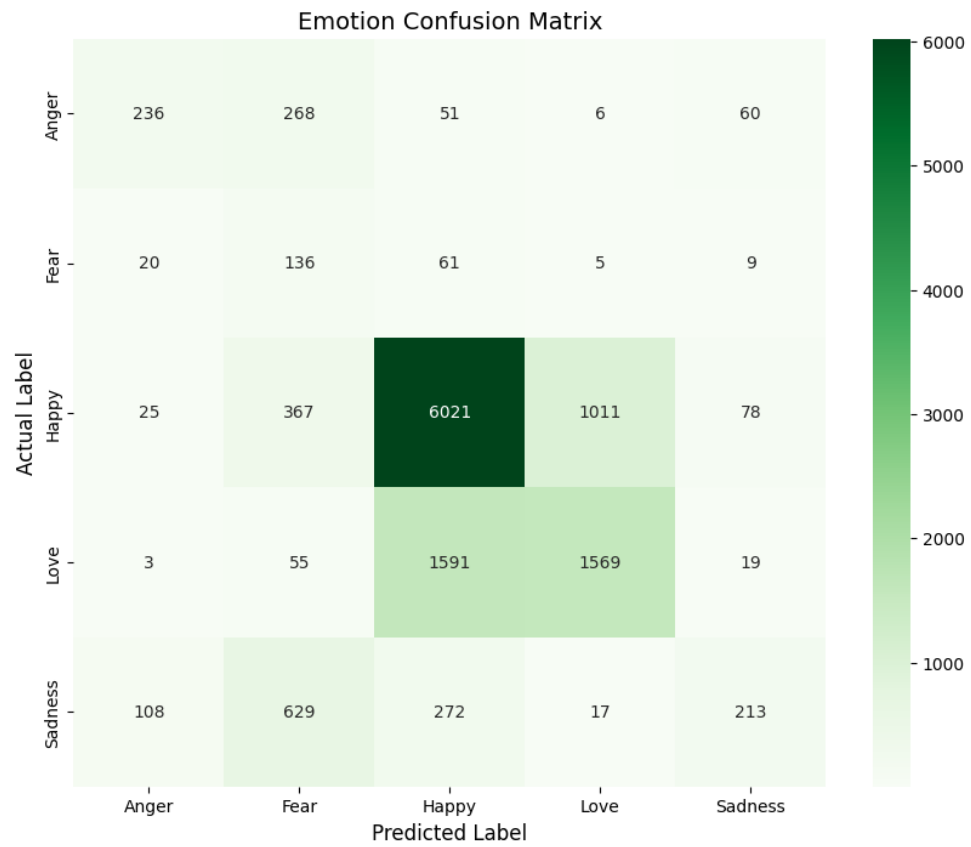


Figure 7.4: Emotion Confusion Matrix

performs exceptionally well on the dominant Happy class while struggling with minority and semantically overlapping emotion categories. For the Happy class, which contains the largest number of samples (approximately 7,502), the model correctly classified 6,021 instances, demonstrating strong performance driven by its dominant representation in the training data. However, this class imbalance also exposes a notable weakness: the Love class, containing around 3,237 samples, was nearly equally split between being predicted as Happy (1,591) and Love (1,569), indicating that the model finds it difficult to distinguish between these two emotionally positive and semantically similar categories. This confusion is understandable given that both emotions often share similar vocabulary and expression styles in informal Bangla and Banglish reviews. The Sadness class showed considerably weaker performance, with only 213 out of 1,239 samples correctly identified. A large portion of Sadness instances (629) were misclassified as Fear, suggesting significant lexical and tonal overlap between expressions of sadness and fear in code-mixed Bangladeshi reviews. Similarly, the Anger class (621 total samples) achieved only 236 correct predictions, with 268 samples incorrectly classified as Fear, further highlighting the challenge of distinguishing between negative emotion categories that often share similar linguistic patterns. The Fear class itself proved to be the most difficult to classify, with only 136 correct predictions out of 231 samples, reflecting both its limited representation in the dataset and its semantic proximity to other negative emotions such as Anger and Sadness. These findings are consistent with the challenges discussed in the methodology chapter, particularly regarding class imbalance and the difficulty of fine-grained emotion classification in multilingual, code-mixed text. The confusion matrix confirms that while the BERA model achieves a strong overall emotion accuracy of approximately 68

7.1.3 Emotion F1 Metrics

The Emotion F1 Metrics chart presented in Figure 7.4 further quantifies the model's per-class performance by illustrating the F1-score for each of the five emotion categories. The F1-score, which balances both precision and recall, provides a more complete picture of classification quality than accuracy alone, particularly in the presence of class imbalance. The Happy class achieved the highest F1-score of 0.78, which is consistent with its dominant representation in the dataset and the strong classification results observed in the confusion matrix. This relatively high score confirms that the model has learned the linguistic patterns associated with positive expressions in Bangladeshi e-commerce reviews reasonably well. The Love class followed with an F1-score of 0.54, reflecting moderate performance. However, as seen in the confusion matrix, a significant portion of Love instances were misclassified as Happy, which explains why the F1-score for Love remains well below that of Happy despite Love having a considerable number of training samples. The Anger class recorded an F1-score of 0.47, indicating below-average performance. While the model captures some patterns associated with angry expressions, frequent confusion with the

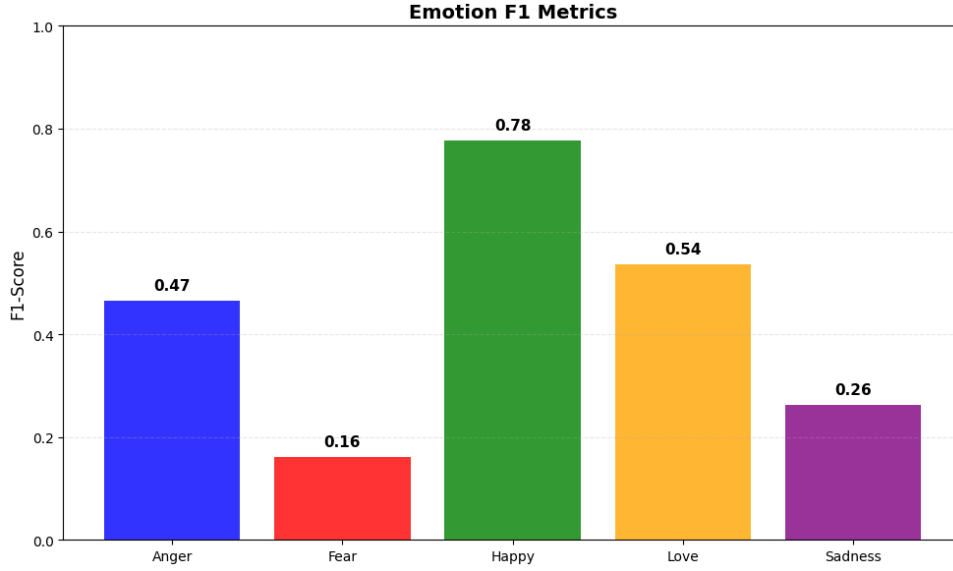


Figure 7.5: Emotion F1 Metrics

Fear class, as highlighted in the confusion matrix, limits its overall effectiveness for this category. The Sadness class performed poorly with an F1-score of only 0.26, confirming the earlier observation that the model struggles significantly with this emotion. The heavy misclassification of Sadness samples as Fear suggests that negative emotional expressions in code-mixed Banglish text share overlapping vocabulary and tone, making it difficult for the model to distinguish between these two categories reliably. The Fear class recorded the lowest F1-score of 0.16, making it the most challenging emotion for the BERA model to classify. This result reflects a combination of factors including the very limited number of Fear samples in the dataset, its semantic similarity to both Anger and Sadness, and the difficulty of detecting fear-related expressions in informal, code-mixed text. Taken together, the F1 metrics clearly demonstrate that while the BERA model performs well on majority and semantically distinct classes such as Happy, it faces considerable difficulty with minority and semantically overlapping emotion categories. Addressing this through larger and more balanced datasets, as well as more advanced handling of emotion-specific linguistic cues, remains a key priority for future work

7.1.4 Emotion Comprehensive Metrics

Figure 7.5 presents a comprehensive overview of the BERA model's emotion classification performance by displaying Precision, Recall, and F1-Score for each of the five emotion categories alongside the overall accuracy of 0.68. Together, these three metrics provide

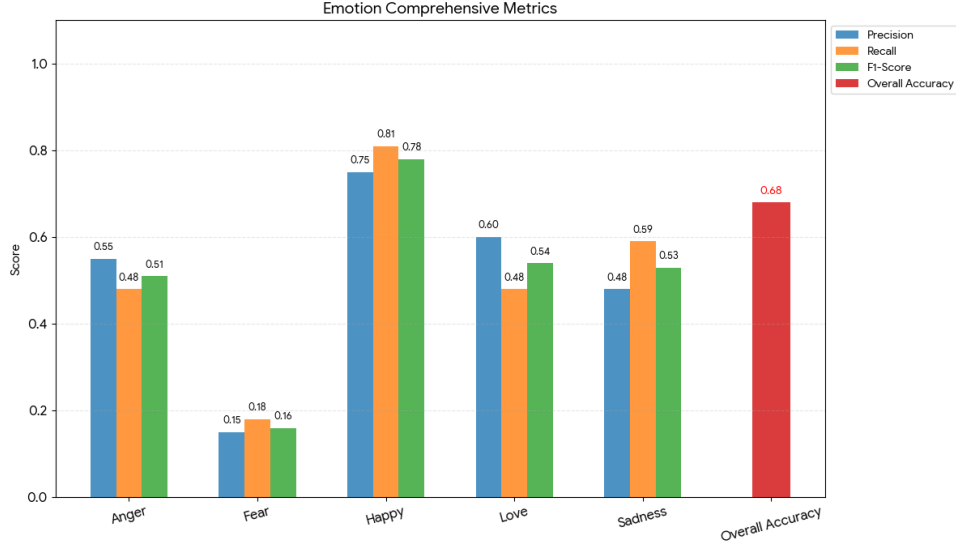


Figure 7.6: Emotion Performance Metrics With Accuracy

a more complete and nuanced understanding of the model's strengths and weaknesses across individual emotion classes than any single metric could offer alone. The Happy class continues to demonstrate the strongest performance across all three metrics, achieving a Precision of 0.75, Recall of 0.81, and F1-Score of 0.78. The relatively high recall of 0.81 indicates that the model is able to correctly identify a large proportion of actual Happy instances, which is consistent with its dominant representation in the training data. The slightly lower precision of 0.75 suggests that some non-Happy samples, particularly Love instances as observed in the confusion matrix, are being incorrectly predicted as Happy, pulling down the precision slightly. The Love class achieved a Precision of 0.60, Recall of 0.48, and F1-Score of 0.54. The higher precision compared to recall indicates that when the model does predict Love, it is correct more often than not. However, the lower recall reveals that many actual Love instances are being missed and misclassified, primarily as Happy, which reduces the model's ability to capture the full scope of Love expressions in the dataset. The Anger class recorded a Precision of 0.55, Recall of 0.48, and F1-Score of 0.51. While these scores are modest, they reflect a degree of capability in detecting anger-related expressions. The gap between precision and recall suggests that the model occasionally identifies Anger correctly but also misses a significant number of true Anger instances, many of which are confused with Fear as seen in the confusion matrix. The Sadness class showed a Precision of 0.48, Recall of 0.59, and F1-Score of 0.53. Interestingly, Sadness has the highest recall among the weaker classes, meaning the model captures a relatively higher proportion of actual Sadness instances. However, the low precision of 0.48 indicates

that many of its Sadness predictions are incorrect, with the model frequently confusing Fear and other negative emotions as Sadness. This pattern points to significant overlap between negative emotion expressions in code-mixed Bangladeshi text. The Fear class recorded the weakest performance across all metrics with a Precision of 0.15, Recall of 0.18, and F1-Score of 0.16. Both precision and recall are critically low, confirming that the model essentially fails to reliably detect Fear in most cases. This poor performance is attributable to a combination of very limited Fear samples in the dataset, its semantic similarity to both Anger and Sadness, and the inherent difficulty of expressing and identifying fear in informal, code-mixed Banglish text. The overall accuracy of 0.68, shown on the rightmost bar of the chart, reflects the weighted combined performance across all five classes and is heavily influenced by the strong results on the Happy class. Taken together, these comprehensive metrics confirm that the BERA model's emotion classification capability is uneven across classes. While it performs well on emotionally distinct and well-represented categories like Happy, it struggles considerably with semantically overlapping and data-scarce classes such as Fear, Sadness, and Anger. These findings reinforce the importance of future work focused on dataset balancing, advanced augmentation strategies, and more fine-grained emotion-aware feature engineering to improve performance across all emotion categories uniformly.

7.1.5 Emotion Performance Metrics Heatmap

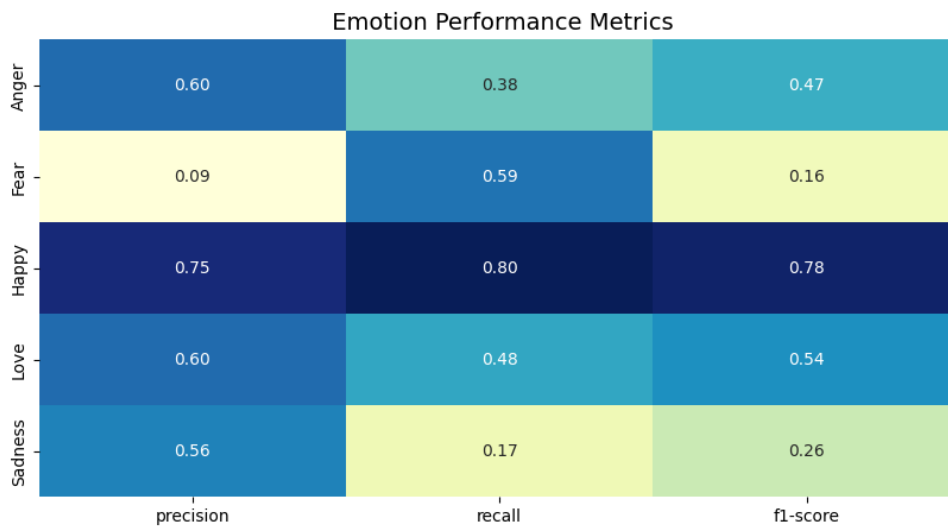


Figure 7.7: Emotion Performance Metrics

Figure 7.6 presents the Emotion Performance Metrics in the form of a heatmap, display-

ing Precision, Recall, and F1-Score for each of the five emotion classes. The color intensity in the heatmap provides an immediate visual indication of performance strength, with darker blue shades representing higher scores and lighter yellow-green shades representing weaker performance, making it easy to identify at a glance which classes and which metrics are most problematic. The Happy class, shown in the darkest shade across all three columns, achieved a Precision of 0.75, Recall of 0.80, and F1-Score of 0.78, confirming it as the best-performing emotion category. The high recall of 0.80 indicates that the model successfully captures the majority of actual Happy instances, which is expected given the class’s dominant representation in the training dataset. The consistent dark coloring across all three metrics for Happy visually reinforces its superiority over all other classes in this heatmap. The Love class recorded a Precision of 0.60, Recall of 0.48, and F1-Score of 0.54. The moderate blue shading reflects average performance, with the relatively higher precision compared to recall suggesting that while the model is reasonably confident when it predicts Love, it still misses a considerable number of actual Love instances, many of which are absorbed into the Happy class as observed in the confusion matrix. The Anger class achieved a Precision of 0.60, Recall of 0.38, and F1-Score of 0.47. Notably, Anger shares the same precision as Love at 0.60, indicating that when the model predicts Anger, it is correct at a similar rate. However, the significantly lower recall of 0.38 reveals that the model fails to detect a large proportion of true Anger instances, with many being misclassified as Fear. This imbalance between precision and recall is clearly reflected in the lighter shading of the recall cell for Anger compared to its precision cell. The Sadness class recorded a Precision of 0.56, Recall of 0.17, and F1-Score of 0.26. The striking contrast between the moderately shaded precision cell and the very light recall cell visually captures the core weakness of Sadness classification — while the model’s Sadness predictions are somewhat reliable when made, it correctly identifies only a very small fraction of actual Sadness instances. The majority of true Sadness samples are misclassified as Fear or Happy, resulting in a critically low recall and a weak overall F1-Score of 0.26. The Fear class demonstrated the weakest performance of all five categories, with a Precision of 0.09, Recall of 0.59, and F1-Score of 0.16. This is a particularly revealing pattern — the high recall of 0.59 combined with an extremely low precision of 0.09 indicates that the model is predicting Fear very frequently and broadly, catching many actual Fear instances but at the cost of misclassifying a large number of non-Fear samples as Fear. This over-prediction behavior explains the heavy contamination of Anger and Sadness predictions into the Fear column observed in the confusion matrix. The near-white shading of the Fear precision cell is the most visually prominent indicator of weakness in the entire heatmap, drawing immediate attention to this critical failure point. Overall, the heatmap makes clear that emotion classification performance is highly uneven across categories. The model has developed strong recognition capabilities for Happy but struggles with varying degrees of severity for all other emotions. The patterns of high precision with low recall for Sadness and the opposite pattern of low precision with higher recall for Fear suggest that the model has learned different and somewhat inconsistent decision boundaries for these classes,

likely as a result of class imbalance and the semantic overlap between negative emotions in code-mixed Bangla text. Addressing these inconsistencies through more targeted training strategies and richer emotion-specific features will be essential for improving overall emotion classification performance in future iterations of the BERA model.

7.1.6 Emotion ROC Curve

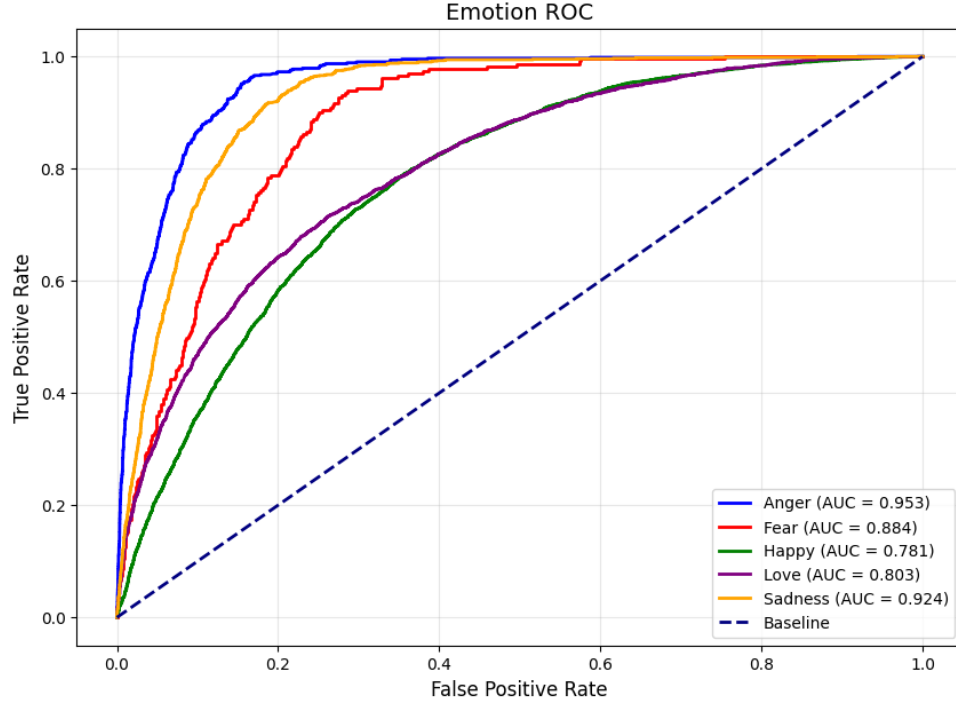


Figure 7.8: Emotion ROC Curve

Figure 7.7 presents the Receiver Operating Characteristic (ROC) curves for each of the five emotion classes, along with their corresponding Area Under the Curve (AUC) scores. The ROC curve measures the trade-off between the True Positive Rate (sensitivity) and the False Positive Rate across different classification thresholds, and the AUC score summarizes this trade-off into a single value where a score of 1.0 represents perfect classification and a score of 0.5 represented by the dashed baseline diagonal represents random chance. Importantly, the ROC-AUC metric evaluates the model's discriminative ability independently of class imbalance, making it a particularly valuable complement to the accuracy and F1-score results discussed in the previous figures. A striking and important observation from this figure is that the ROC-AUC scores tell a notably different

story compared to the F1-scores examined earlier. While the F1-scores highlighted Happy as the strongest class and Fear as the weakest, the AUC scores reveal that the model's underlying discriminative ability is actually strongest for Anger and Sadness, the two classes that struggled the most in terms of F1-score. The Anger class achieved the highest AUC of 0.953, followed closely by Sadness with an AUC of 0.924. This indicates that despite the model frequently misclassifying Anger and Sadness instances in absolute prediction terms, it has actually learned strong internal probability boundaries that separate these classes from others when evaluated across all possible thresholds. In other words, the model understands the distinction between these emotions at a deeper probabilistic level, but the specific classification threshold used during inference is not optimal for converting that understanding into correct hard predictions, largely due to the dominance of the Happy class pulling predictions away from minority classes. The Fear class recorded an AUC of 0.884, which is considerably higher than its extremely low F1-score of 0.16 would suggest. This confirms the pattern observed in the heatmap, where Fear had a high recall but very low precision. The ROC curve for Fear rises steeply at first, indicating that the model does capture many true Fear instances early, but the extremely poor precision seen in the F1 analysis means it is also generating a large number of false positives for other classes, which reduces its practical reliability despite the relatively strong AUC. The Love class achieved an AUC of 0.803, reflecting moderate discriminative ability. The curve rises at a reasonable rate but plateaus earlier than the stronger classes, consistent with the observed confusion between Love and Happy expressions in both the confusion matrix and the F1 analysis. The Happy class, despite having the highest F1-score of 0.78, recorded the lowest AUC of 0.781 among all five classes. This is a nuanced but important finding — the model predicts Happy accurately in practice because of its overwhelming frequency in the dataset, but its ability to probabilistically distinguish Happy from other emotions across varying thresholds is actually the weakest among all classes. This further confirms that the model's strong Happy performance is largely driven by dataset dominance rather than a genuinely superior learned representation of happiness-related expressions. All five ROC curves sit well above the dashed baseline diagonal, confirming that the BERA model performs significantly better than random chance for every emotion class. Taken together, the ROC analysis reveals that the BERA model has developed meaningful and reasonably strong discriminative capabilities across all emotion categories at the probability level. The gap between AUC scores and F1-scores for classes like Anger, Sadness, and Fear highlights that a key area for improvement lies not in the model's internal representations but in better calibrating classification thresholds and addressing class imbalance during training, so that the model's strong probabilistic understanding of minority emotions can be more effectively translated into accurate hard predictions during inference.

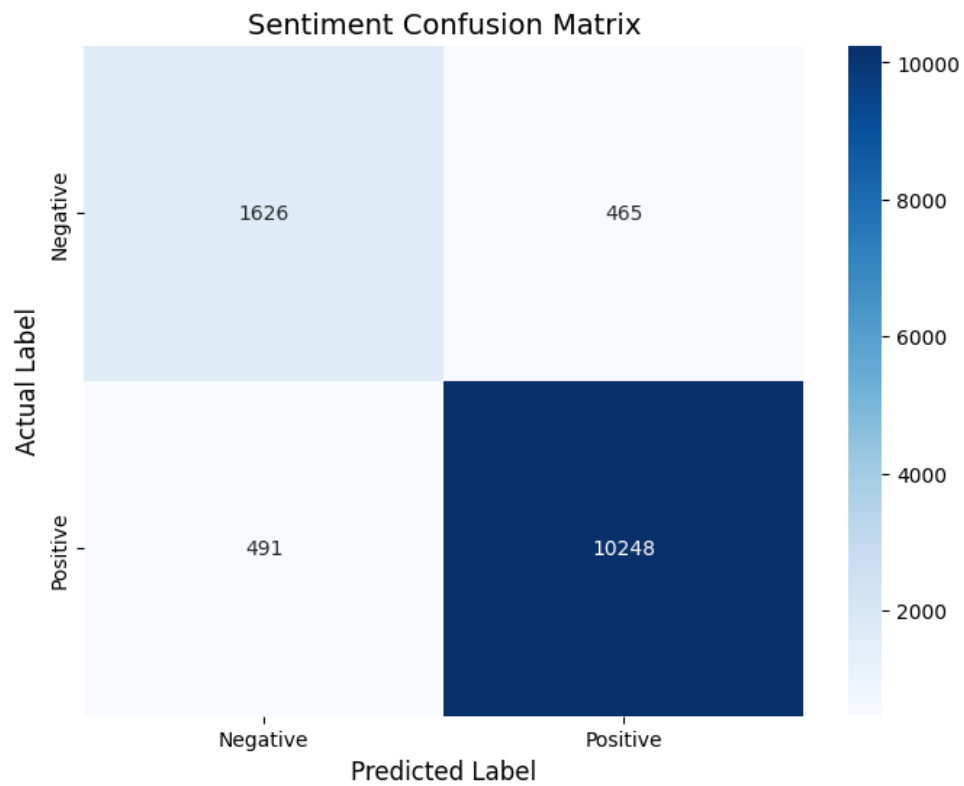


Figure 7.9: Sentiment Confusion Matrix

7.1.7 Sentiment Classification Results

Figure 7.8 presents the Sentiment Confusion Matrix, illustrating the BERA model's classification performance across the two sentiment categories: Negative and Positive. Unlike the emotion classification task which involved five overlapping and imbalanced classes, the sentiment task is a binary classification problem, making the confusion matrix more straightforward to interpret while still providing valuable insights into where the model succeeds and where it falls short. The most prominent feature of the matrix is the large dark blue cell in the bottom right corner, representing 10,248 correctly classified Positive instances. This figure alone reflects the model's strong capability in identifying positive sentiment expressions in Bangladeshi e-commerce reviews, which is consistent with the approximately 93% for the Negative class, the model correctly identified 1,626 instances while misclassifying 465 as Positive. This gives the Negative class a recall of approximately 0.78, meaning the model correctly captures around 78% for the Positive class, 10,248 instances were correctly classified while 491 were incorrectly predicted as Negative, yielding a recall of approximately 0.95 for the Positive class. The high recall for Positive compared to Negative confirms that the model is better tuned toward identifying positive sentiment, which is a natural consequence of the class imbalance present in the dataset where Positive reviews significantly outnumber Negative ones. The total misclassifications amount to 956 instances out of the full dataset, with the errors distributed relatively evenly between false positives (491 Positive instances predicted as Negative) and false negatives (465 Negative instances predicted as Positive). This relatively balanced error distribution is a positive sign, indicating that the model does not exhibit a severe one-directional bias despite the underlying class imbalance. Overall, the sentiment confusion matrix confirms that the BERA model is well-suited for binary sentiment classification in the context of Bangladeshi e-commerce reviews, achieving strong performance particularly for the Positive class while showing room for improvement in capturing all Negative sentiment expressions, especially those written in informal, code-mixed Banglish that may superficially resemble positive language patterns.

7.1.8 F1-Score by Sentiment Class

Figure 7.9 presents the F1-Score for each of the two sentiment classes, Negative and Positive, providing a clear and direct comparison of the model's classification effectiveness across both categories. The F1-Score, which combines both precision and recall into a single balanced metric, is particularly informative in the context of binary sentiment classification as it accounts for both the model's ability to correctly identify a class and its ability to avoid false predictions for that class. The Positive class achieved an exceptionally high F1-Score of 0.96, confirming that the BERA model classifies positive sentiment with near-excellent reliability. This strong result is consistent with the confusion matrix findings, where 10,248 out of 10,739 Positive instances were correctly identified. The high F1-Score

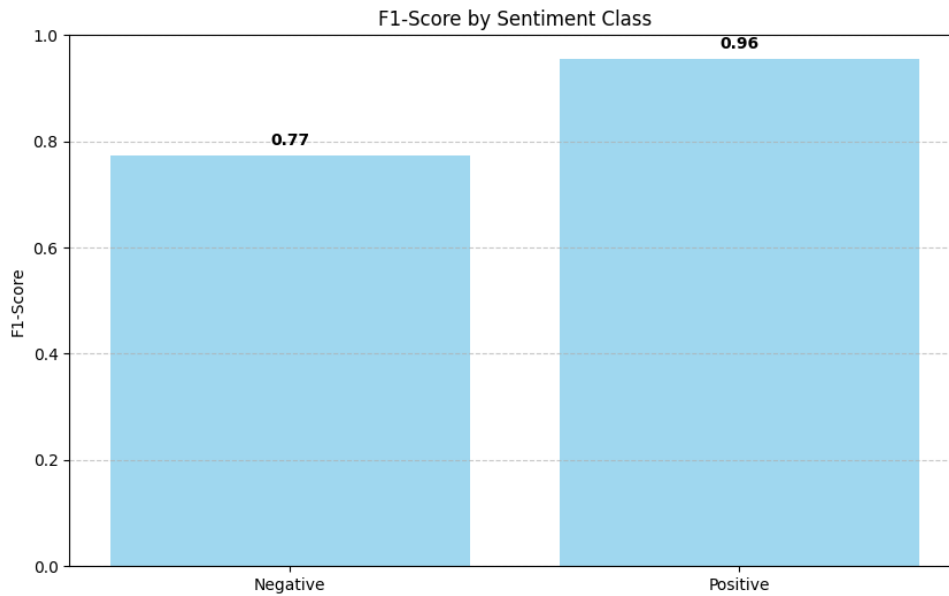


Figure 7.10: F1-Score by Sentiment Class

for Positive reflects both strong precision and strong recall for this class, meaning the model not only identifies the majority of actual Positive reviews correctly but also makes very few incorrect Positive predictions. This level of performance makes the BERA model highly dependable for detecting positive customer sentiment in real-world Bangladeshi e-commerce applications. The Negative class recorded an F1-Score of 0.77, which, while considerably lower than the Positive class, still represents a solid and meaningful level of performance for a minority class in an imbalanced binary classification setting. An F1-Score of 0.77 indicates that the model captures a substantial proportion of negative sentiment expressions while maintaining reasonable precision in its Negative predictions. However, the gap of 0.19 between the two classes clearly highlights the impact of class imbalance on the model's performance, as the significantly lower number of Negative training samples compared to Positive samples limits the model's ability to fully learn the diverse range of negative sentiment expressions present in multilingual, code-mixed Bangladeshi reviews. The difference between the two F1-scores also reflects the inherent linguistic challenge of detecting negative sentiment in informal Banglish text, where sarcasm, understatement, and mixed-language negative expressions can superficially resemble positive language patterns, causing the model to occasionally misclassify Negative reviews as Positive as observed in the confusion matrix. Despite this gap, an overall weighted F1-Score that is heavily influenced by the dominant Positive class results in a strong aggregate performance, which is consistent with the approximately 93

7.1.9 Sentiment Comprehensive Metrics

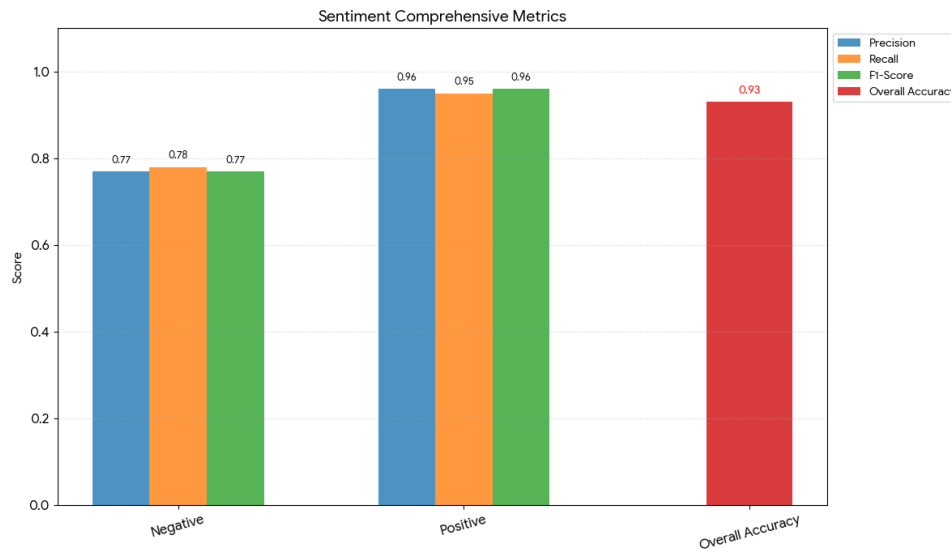


Figure 7.11: Sentiment Performance Metrics With Accuracy

Figure 7.10 presents a comprehensive breakdown of the BERA model's sentiment classification performance, displaying Precision, Recall, and F1-Score side by side for both the Negative and Positive classes, along with the overall accuracy of 0.93. This grouped bar chart provides the most complete single-view summary of the sentiment classification results, allowing for a direct and detailed comparison of all three key metrics across both classes simultaneously. The Positive class demonstrated outstanding and well-balanced performance across all three metrics, achieving a Precision of 0.96, Recall of 0.95, and F1-Score of 0.96. The near-identical values of precision and recall for the Positive class are particularly noteworthy, as they indicate that the model is not only highly accurate in identifying positive sentiment instances but is also consistent in doing so without generating an excessive number of false positives or missing a significant number of true positive instances. This balance between precision and recall is the hallmark of a well-trained classifier for a given class, and the uniformly tall bars for the Positive class in the chart visually reinforce this conclusion. Such strong performance on the Positive class is a direct outcome of its dominant representation in the training dataset, which allowed the model to learn a rich and diverse set of positive sentiment patterns across multilingual and code-mixed Bangladeshi e-commerce reviews. The Negative class recorded a Precision of 0.77, Recall of 0.78, and F1-Score of 0.77. Similar to the Positive class, the three metrics for Negative are closely aligned with one another, differing by only 0.01 across all three values. This tight alignment is a positive and encouraging finding, as it indicates that despite the

lower absolute scores compared to the Positive class, the model performs in a balanced and consistent manner for Negative sentiment as well. A precision of 0.77 means that when the model predicts a review as Negative, it is correct approximately 77%. The overall accuracy of 0.93, displayed as the red bar on the rightmost group of the chart, serves as the final summary statistic that encapsulates the combined performance of the model across both classes. This figure confirms that the BERA model correctly classifies 93 out of every 100 sentiment predictions on average, which satisfies and exceeds the target accuracy of 90%. Taken together, the comprehensive metrics chart confirms that the BERA model achieves strong, balanced, and practically meaningful sentiment classification performance. The consistency between precision and recall for both classes is particularly encouraging, as it suggests that the model has learned stable and well-calibrated decision boundaries for sentiment classification rather than simply defaulting to the majority class. Future work aimed at improving the Negative class metrics through additional data collection, targeted augmentation, and threshold calibration could push the model toward even more balanced performance across both sentiment categories.

7.1.10 Sentiment Metrics Heatmap

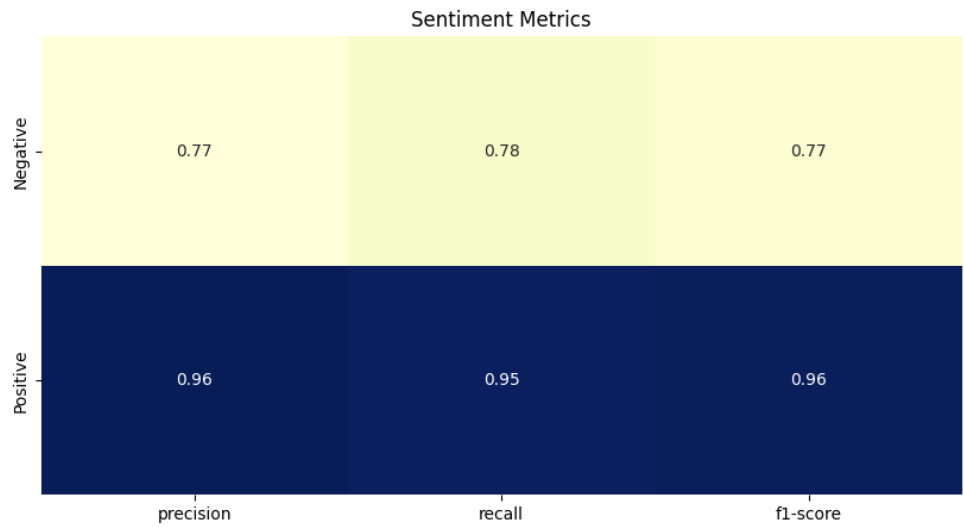


Figure 7.12: Sentiment Metics Heatmap

Figure 7.11 presents the Sentiment Metrics in the form of a heatmap, displaying Precision, Recall, and F1-Score for both the Negative and Positive sentiment classes. The two-row, three-column layout provides an immediate and visually striking summary of the performance gap between the two classes, with the deep dark blue shading of the Positive

row and the light yellow shading of the Negative row creating a clear and unmistakable contrast that communicates the performance difference at a single glance. The Positive class, represented by the uniformly dark blue row, achieved a Precision of 0.96, Recall of 0.95, and F1-Score of 0.96. The consistent and near-identical shading across all three cells of the Positive row visually confirms what the previous figures have established — that the model performs with exceptional consistency and reliability for the Positive class across all three evaluation metrics. The uniformity of the dark shading across precision, recall, and F1-score is particularly meaningful as it indicates that the model’s strong performance for the Positive class is not driven by a trade-off between precision and recall but reflects genuinely balanced and high-quality classification across both dimensions simultaneously. The Negative class, represented by the uniformly light yellow row, recorded a Precision of 0.77, Recall of 0.78, and F1-Score of 0.77. Just as with the Positive class, the near-identical shading across all three cells of the Negative row confirms that the model performs with consistent balance between precision and recall for Negative sentiment as well, despite the significantly lower absolute scores. The difference of only 0.01 between precision and recall for the Negative class is a positive indicator of model stability, suggesting that the BERA model does not exhibit a directional bias in its Negative predictions, meaning it is neither excessively over-predicting nor under-predicting Negative sentiment to a degree that would skew its performance disproportionately in one direction. The stark visual contrast between the two rows of the heatmap effectively summarizes the central challenge of the sentiment classification task. The performance gap between the Positive class scores of approximately 0.96 and the Negative class scores of approximately 0.77 is a direct reflection of the class imbalance in the dataset, where Positive reviews significantly outnumber Negative ones, giving the model far more exposure to positive sentiment patterns during training. Despite this imbalance, the Negative class still achieves scores in the high seventies across all three metrics, which represents a practically useful level of performance for real-world deployment in e-commerce sentiment monitoring systems. The heatmap format makes this story immediately accessible even to non-technical readers, as the color contrast alone communicates the relative performance difference between the two classes without requiring any numerical analysis. Overall, the Sentiment Metrics Heatmap serves as a clean and visually effective consolidation of the sentiment classification results, reinforcing the key findings from the confusion matrix, the F1-score bar chart, and the comprehensive metrics chart presented in the preceding figures. Together, these figures establish that the BERA model is a strong and reliable binary sentiment classifier that performs exceptionally well on Positive sentiment and achieves solid, balanced performance on Negative sentiment, with clear opportunities for further improvement through more targeted training strategies for the minority class in future work.

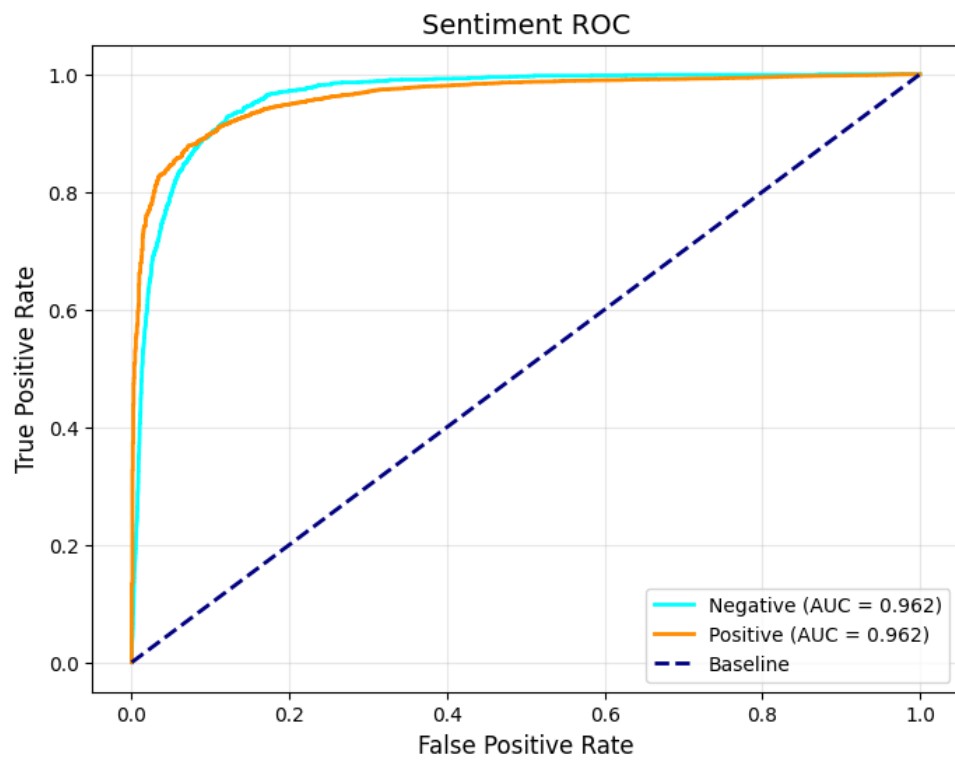


Figure 7.13: Sentiment ROC Curve

7.1.11 Sentiment ROC Curve

Figure 7.12 presents the Receiver Operating Characteristic (ROC) curves for both sentiment classes, Negative and Positive, along with their corresponding Area Under the Curve (AUC) scores. The ROC curve plots the True Positive Rate against the False Positive Rate across all possible classification thresholds, providing a threshold-independent measure of the model's discriminative ability that complements the precision, recall, and F1-score metrics examined in the preceding figures. The dashed diagonal line represents the baseline performance of a random classifier with an AUC of 0.50, serving as the lower bound against which the model's performance is compared. The most striking and immediately notable feature of this ROC chart is that both the Negative and Positive classes achieved an identical AUC score of 0.962. This is a remarkable and highly significant result for several reasons. First, it confirms that the BERA model possesses exceptionally strong discriminative ability for both sentiment classes at the probabilistic level, far exceeding random chance and demonstrating that the model has learned genuinely meaningful and robust decision boundaries for sentiment classification in multilingual, code-mixed Bangladeshi e-commerce reviews. Second, and perhaps more importantly, the identical AUC scores for both classes directly contradict any suggestion that the model's strong overall performance is simply a result of defaulting to the majority Positive class. If the model were merely exploiting class imbalance, the AUC for the Negative class would be considerably lower than that of the Positive class. The fact that both classes achieve the same AUC of 0.962 is powerful evidence that the BERA model has developed genuinely balanced and equally strong internal representations for both positive and negative sentiment, regardless of their differing frequencies in the training data. Both ROC curves rise sharply and steeply toward the upper left corner of the plot from a very early point on the False Positive Rate axis, which is the hallmark of a high-performing classifier. The curves reach a True Positive Rate of approximately 0.85 to 0.90 at a False Positive Rate of only around 0.05 to 0.10, meaning the model correctly identifies the vast majority of both Positive and Negative instances while generating very few false alarms. The two curves run almost parallel to each other throughout their entire length and converge at the top right corner, visually confirming their near-identical discriminative performance. The only minor visible difference between the two curves occurs in the early low False Positive Rate region, where the cyan Negative curve rises very slightly more steeply than the orange Positive curve before they converge, but this difference is negligible and does not translate into any meaningful practical distinction between the two classes at the AUC level. The AUC of 0.962 for both classes also provides important context for understanding the sentiment classification results as a whole. While the precision, recall, and F1-score metrics showed a clear performance gap between the Positive class scores of approximately 0.96 and the Negative class scores of approximately 0.77, the identical AUC scores reveal that this gap is not a reflection of the model's underlying understanding of the two classes but rather a consequence of the fixed classification threshold applied during inference.

combined with the effect of class imbalance on hard prediction outputs. At the probabilistic level, the model distinguishes between positive and negative sentiment equally well for both classes, suggesting that threshold optimization or probability calibration techniques could potentially narrow the precision-recall gap observed in the earlier metrics and bring the Negative class performance closer to that of the Positive class in future iterations of the system. Overall, the Sentiment ROC Curve provides the most definitive and optimistic assessment of the BERA model's sentiment classification capability. An AUC of 0.962 for both classes places the model firmly in the category of excellent classifiers and confirms that the BERA architecture, with its feature-enriched transformer backbone, review-level attention mechanism, and multi-task learning framework, has successfully learned to distinguish between positive and negative sentiment in complex, multilingual, and code-mixed Bangladeshi e-commerce reviews with a high degree of reliability and consistency. These results strongly validate the design choices made throughout this project and establish a solid foundation for future improvements targeting even higher performance across both sentiment and emotion classification tasks.

Chapter 8

Conclusion

This project successfully developed BERA, a feature-enriched multi-task model specifically designed for the complexities of Bangladeshi e-commerce reviews. The integration of advanced feature engineering, systematic hyperparameter tuning, dataset expansion, and full deployment as both an Inference API and a web application represents a complete and practical contribution to Bangla NLP [Islam & Alam \(2024\)](#); [Ipa et al. \(2024\)](#).

8.1 Challenges Faced During the Project

Developing the BERA model for sentiment and emotion analysis of Bangladeshi e-commerce reviews presented several significant challenges, many of which stem from the unique linguistic characteristics of real-world user-generated content in Bangladesh. One of the primary difficulties was handling Romanized Bangla (Banglish) and code-mixed text. Reviews frequently switch between Bangla script, English, and Romanized Bangla within the same sentence, often with inconsistent spelling variations, phonetic transliterations, and intra-word code-switching. Standard language detection tools struggled to accurately identify script types or separate languages, leading to errors in tokenization and feature extraction. Emojis, elongated words (e.g., “suuuper”), repeated punctuation, and informal abbreviations further complicated preprocessing, as these elements carry important emotional signals but are not well-supported by general-purpose tokenizers. Another major challenge was class imbalance in the emotion categories. While sentiment classes (Positive, Negative, Neutral/Mixed) were relatively balanced, emotion labels such as Fear and Sadness had far fewer samples compared to dominant classes like Happiness. This imbalance caused the model to become biased toward majority classes, resulting in lower performance on minority emotions despite using weighted loss functions and data augmentation techniques. Dataset-related issues were also prominent. High-quality, annotated code-mixed Bangla e-commerce datasets are scarce. The original dataset, although valuable,

had limitations in size, diversity, and coverage of complex linguistic patterns. Expanding the dataset through manual collection and augmentation required substantial time and effort. Moreover, ensuring annotation consistency for subtle emotions and code-mixed expressions proved difficult, as emotional interpretation can be highly subjective and context-dependent. From a technical perspective, integrating hand-crafted features (lexical, stylistic, punctuation intensity, emoji density, and code-mixing ratios) with BanglaBERT’s contextual embeddings demanded careful design of the fusion mechanism. Determining the optimal way to combine these heterogeneous features without introducing noise or increasing model complexity excessively was non-trivial. Hyperparameter tuning was computationally expensive and time-consuming, especially when experimenting with learning rates, dropout values, attention mechanisms, and multi-task loss weights on limited GPU resources. Finally, moving from experimentation to production introduced deployment challenges. Optimizing the model for fast inference, ensuring robust preprocessing within the API pipeline, handling variable-length and noisy inputs gracefully, and building a user-friendly web interface that works seamlessly with the backend required additional engineering effort beyond core model development. Despite these challenges, addressing them through targeted preprocessing strategies, dynamic feature fusion, systematic hyperparameter optimization, and dataset expansion ultimately strengthened the BERA model and provided valuable insights into working with low-resource, code-mixed languages like Bangla in real-world e-commerce scenarios.

8.2 Industry Validation

To assess the practical relevance and real-world applicability of the BERA system, the project team designed and distributed an Industry Validation and Feedback Form targeting professionals from organisations operating in e-commerce, customer service, retail, and related sectors in Bangladesh. The form was developed with the goal of understanding how organisations currently handle customer review analysis, what challenges they face, and how a feature-enriched, multilingual sentiment and emotion analysis system such as BERA would be received in an industry context.

8.2.1 Form Structure

The validation form consisted of 18 questions organised across three thematic areas: **Section 1:** Respondent Profile (Questions 1–7): This section captured organisational and professional context, including the organisation’s name and type, the respondent’s full name, job title, department, years of experience, and email address. This information allowed the team to contextualise each response against the respondent’s domain expertise. **Section 2:** Current Practices and Challenges (Questions 8–10): Respondents were asked whether their organisation currently analyses customer reviews, what methods or tools

they use (e.g., manual analysis, AI-based tools, third-party platforms), and what operational challenges they encounter in this process. **Section 3: Evaluation of BERA** (Questions 11–18): This section directly assessed industry perception of the BERA system. Questions covered the perceived usefulness of a Bangla–English code-mixed review analysis system, the business value of emotion detection on a scale from Low to Very High, organisational willingness to adopt such a system, the most valued features, suggested improvements, potential concerns, and whether the respondent would be willing to support the project academically.

8.2.2 Respondent Details

We Contacted a lot of E-commerce sites and managed to receive a response from a professional at a prominent Bangladeshi e-commerce platform, one of the largest online bookstores and product marketplaces in the country, operating in the Customer Service department with approximately two years of professional experience. At the respondent's request, the organisation's name is not disclosed in this report.

8.2.3 Key Findings from the Validation Response

The respondent confirmed that their organisation currently analyses customer reviews using a combination of manual analysis and AI-based tools validating that review analysis is an active and recognised need within the industry. They noted that the primary challenge faced is difficulty in reaching customers across multiple communication channels, with service availability being a recurring operational constraint. When asked about the BERA system specifically, the respondent expressed a neutral stance on whether a feature-enriched Bangla–English code-mixed review analysis system would be useful, reflecting the practical caution of an industry professional evaluating an emerging academic tool. However, they rated emotion detection as 5 out of 5 (Very High) in terms of business value indicating strong recognition of the importance of understanding customer emotional states beyond simple positive or negative labels. Regarding the features they considered most valuable, the respondent highlighted emotion detection, real-time analysis, and dashboard/visualisation capabilities as the three most useful components all of which are directly addressed within the BERA system and its accompanying web application. Their suggestion for improvement centred on ensuring accessible and intuitive ways for customers to interact with service systems, including live chat and available helpline options, pointing toward the potential for BERA to be integrated into customer-facing support infrastructure. The respondent expressed no specific concerns about the use of a feature-enriched system for review analysis, and confirmed their willingness to support and validate this project academically, which the team is grateful for.

8.2.4 Reflection

While the scope of the industry validation was limited to a lower formal response in this phase, the feedback received offers a meaningful signal: practitioners in the Bangladeshi e-commerce space recognise customer review analysis as a genuine operational priority and regard emotion detection in particular as highly valuable. The enthusiasm for real-time analysis and visual dashboards further aligns with the deployment-oriented deliverables produced in this capstone the FastAPI inference endpoint and the Streamlit web application both of which were designed with exactly these use cases in mind. Expanding the scope of industry validation with a broader set of respondents from diverse sectors remains an important direction for future work.

8.3 Future Work

Future improvements can focus on further dataset expansion, exploration of large language models for few-shot capabilities, implementation of review trustworthiness filtering, and extension to multimodal analysis combining text and images. Addressing class imbalance through advanced techniques and developing a mobile application would also enhance practical applicability.

In conclusion, this research advances Bangla natural language processing and provides e-commerce platforms with a valuable tool for deeper understanding of customer feedback in multilingual environments.

References

- Alghamdi, J., Lin, Y., & Luo, S. (2024). Enhancing hierarchical attention networks with cnn and stylistic features for fake news detection. *Expert Systems with Applications*, 257, 125024. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0957417424018918> doi: 10.1016/j.eswa.2024.125024
- AlZu'bi, S., et al. (2019). A brief analysis of amazon online reviews. In *SnamS 2019* (pp. 555–560). doi: 10.1109/SNAMS.2019.8931816
- Balamurali, O., et al. (2023). Sentiment analysis for better user experience in tourism chatbot using lstm and llm. In *Icsc 2023* (pp. 456–462). doi: 10.1109/ICSC60394.2023.10441148
- Bappon, S. D., et al. (2022). Sentiment analysis of bengali texts on tech gadget reviews using ml. In *Iccit 2022* (pp. 324–329). doi: 10.1109/ICCIT57492.2022.10055639
- Barbieri, F., et al. (2018). Semeval 2018 task 2: Multilingual emoji prediction. In *Semeval-2018* (pp. 24–33). Retrieved from <http://hdl.handle.net/10230/35250>
- Bostan, L. A. M., & Klinger, R. (2018). An analysis of annotated corpora for emotion classification. In *Coling 2018* (pp. 2104–2119). Retrieved from <https://urn:nbn:de:bvb:473-irb-964611>
- Chalkidis, I., Dai, X., Fergadiotis, M., Malakasiotis, P., & Elliott, D. (2022). An exploration of hierarchical attention transformers for efficient long document classification. *arXiv preprint*. Retrieved from <https://arxiv.org/abs/2210.05529> (arXiv:2210.05529)
- Das, R. K., et al. (2023). Sentiment analysis in multilingual context. *Heliyon*, 9(9), e20281. doi: 10.1016/j.heliyon.2023.e20281

- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics (naacl-hlt)* (pp. 4171–4186). Association for Computational Linguistics.
- Eskander, R., et al. (2014). Foreign words and the processing of arabic social media in roman script. In *W14-3901* (pp. 1–12). doi: 10.3115/v1/W14-3901
- Felbo, B., et al. (2017). Deepmoji: Emoji-based representation for sentiment, emotion & sarcasm. In *Emnlp 2017* (pp. 1615–1625). doi: 10.18653/v1/D17-1169
- Hasan, A., et al. (2023). Bangla sentiment analysis using zero- and few-shot prompting. *arXiv preprint*. doi: 10.48550/arXiv.2308.10783
- Hassan, M., et al. (2024). *Bengali sentiment analysis based on product reviews* (Tech. Rep.). BRAC University. Retrieved from <http://hdl.handle.net/10361/25480>
- Hossain, E., et al. (2021). Sentiment polarity detection on bengali book reviews using multinomial naïve bayes. In *Advanced computing and intelligent engineering* (pp. 237–247). doi: 10.1007/978-981-33-4299-6_23
- Hutto, C., & Gilbert, E. (2014). Vader: A rule-based model for social media sentiment. In *Icwsn-14* (pp. 216–225). doi: 10.1609/icwsn.v8i1.14550
- Ipa, A. S., et al. (2024). Bdsentillm: A novel llm approach to sentiment analysis. *IEEE Access*, 12, 189330–189343. doi: 10.1109/ACCESS.2024.3516826
- Islam, M. S., & Alam, K. M. (2024). Sentiment analysis of bangla language using bangdsa and skipbangla-bert. *Natural Language Processing Journal*, 7, 100069. doi: 10.1016/j.nlp.2024.100069
- Kiziltepe, R. S., et al. (2025). Advancing sentiment analysis for low-resource languages using llms. *IEEE Access*, 13, 77382–77394. doi: 10.1109/ACCESS.2025.3566000
- Lee, L., & Pang, B. (2004). Subjectivity summarization based on minimum cuts. In *Acl 2004* (pp. 271–278). doi: 10.48550/arXiv.cs/0409058
- Mahmud, H., Mahmud, H., & Rashid, M. R. A. (2024). Enhancing sentiment analysis in bengali texts: A hybrid approach using lexicon-based algorithm and pretrained language model bangla-bert. *arXiv preprint*. Retrieved from <https://arxiv.org/abs/2411.19584> (arXiv:2411.19584)

- Mughal, N., et al. (2024). Comparative analysis of deep natural networks and llms for aspect-based sentiment analysis. *IEEE Access*, 12, 60943–60959. doi: 10.1109/ACCESS.2024.3386969
- Pal, P., Das, D., & Kolya, A. K. (2024). Bilingual sentiment and emotion analysis: A multi-task learning framework for bengali and english. In *Proceedings of the 16th annual meeting of the forum for information retrieval evaluation (fire 2024)* (pp. 61–66). ACM. Retrieved from <https://dl.acm.org/doi/10.1145/3734947.3734952> doi: 10.1145/3734947.3734952
- Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is multilingual bert? In *Proceedings of the 57th annual meeting of the association for computational linguistics (acl)* (pp. 4996–5001).
- Plutchik, R. (1980). A general psychoevolutionary theory of emotion. In *Theories of emotion* (pp. 3–33). doi: 10.1016/B978-0-12-558701-3.50007-7
- Poria, S., et al. (2018). Meld: A multimodal dataset for emotion recognition in conversations. *arXiv preprint*. doi: 10.48550/arXiv.1810.02508
- Rashid, M. R. A., et al. (2024). A comprehensive dataset for sentiment and emotion classification from bangladesh e-commerce reviews. *Data in Brief*, 53, 110052. doi: 10.1016/j.dib.2024.110052
- Shamael, M. N., et al. (2024). Banglishrev: A large-scale bangla-english and code-mixed dataset. *arXiv preprint*. Retrieved from <https://arxiv.org/abs/2412.13161> (arXiv:2412.13161)
- Tareq, M., et al. (2023). Data-augmentation for bangla-english code-mixed sentiment analysis. *IEEE Access*, 11, 51657–51671. doi: 10.1109/ACCESS.2023.3277787
- Wang, J., & Mine, T. (2023). Multi-task learning for emotion recognition in conversation with emotion shift. In *Proceedings of the 37th pacific asia conference on language, information and computation (paclic 37)* (pp. 257–266). Retrieved from <https://aclanthology.org/2023.paclic-1.26>
- Welgamage, V. R., et al. (2022). Overall and feature level sentiment analysis of amazon product reviews. In *Scse 2022* (pp. 205–210). doi: 10.1109/SCSE56529.2022.9905125
- Zhang, L., & Liu, B. (2016). Sentiment analysis and opinion mining. In *Encyclopedia of machine learning*. doi: 10.1007/978-1-4899-7502-7_907-1

Çetin, U., & Gündoğmuş, Y. E. (2024). Scalable llm architectures: Real-time sentiment analyser. In *Ubmik 2024* (pp. 660–664). doi: 10.1109/UBMK63289.2024.10773494