

Lightweight Transformer-Based Ai Model for Multilingual and Code-Mixed Sentiment Analysis

Mr. Vijay Kumar

ABSTRACT

Social media users increasingly communicate across language boundaries, often combining two or more languages within the same sentence and frequently writing Indian languages in Roman script. This creates a difficult setting for sentiment analysis because conventional monolingual models must cope with spelling variation, transliteration, informal vocabulary, emojis, abbreviations, language switching, and context-dependent sentiment. Large multilingual Transformer models have improved contextual understanding, but their computational cost can make deployment difficult on resource-constrained systems. This thesis proposes a lightweight Transformer-based framework for multilingual and code-mixed sentiment analysis. The framework uses a multilingual teacher model, language-aware normalization, and knowledge distillation to transfer contextual knowledge into a smaller student model. The study focuses primarily on Hindi-English (Hinglish) and Spanish-English (Spanglish) social-media text, using the SemEval-2020 Task 9 SentiMix corpora as a reproducible benchmark. The proposed framework is evaluated against classical machine-learning, recurrent neural-network, and full-size Transformer baselines. Evaluation is designed to consider not only accuracy and macro/weighted F1, but also inference latency, parameter count, memory consumption, robustness to noisy and code-mixed text, and the consistency of predictions across language pairs. An explainability component is also included so that important tokens contributing to a prediction can be inspected. The central research question is whether a substantially smaller Transformer can retain useful multilingual sentiment knowledge while reducing computational requirements. The thesis therefore treats sentiment analysis as both a language-understanding problem and an engineering deployment problem. The resulting framework is intended for applications such as social-media monitoring, customer-feedback analysis, multilingual communication systems, and other AI systems where language diversity and limited computing resources must be considered together.

Keywords: Transformer, multilingual NLP, code-mixed text, sentiment analysis, knowledge distillation, Hinglish, Spanglish, lightweight AI, explainable AI, social media.

CHAPTER 1 – INTRODUCTION

1.1 Background

The growth of social-media platforms has changed the way people express opinions. A short post can contain a mixture of languages, transliterated words, slang, abbreviations, emojis, hashtags, and references that depend heavily on context. For a human reader, these variations are often manageable because people use cultural knowledge and conversational context. For an AI model, however, the same variation can turn a simple sentiment classification problem into a difficult language-understanding task.

Sentiment analysis attempts to determine the emotional or evaluative orientation of a text, commonly using positive, negative, and neutral classes. Early systems often depended on sentiment dictionaries or manually designed features. Machine-learning models then learned statistical patterns from labelled examples, followed by deep neural networks that learned representations directly from text. Transformer architectures changed the field by using attention to model relationships among tokens across a sequence. BERT demonstrated that contextual bidirectional representations could be fine-tuned for many language tasks with a relatively simple task-specific output layer [1].

The success of Transformers has created a new engineering problem. The strongest multilingual models are often larger than what is convenient for low-latency or resource-limited deployment. DistilBERT demonstrated that knowledge distillation can create a smaller and faster model while retaining much of the capability of a larger teacher [2]. TinyBERT further showed that Transformer-specific distillation can transfer knowledge from a large teacher to a smaller student [3]. These ideas motivate the present work.

1.2 Multilingual and Code-Mixed Text

Multilingual text contains more than one language across a dataset or application. Code-mixed text is more demanding because language switching can occur within the same sentence or even within a single phrase. In Indian social media, Hindi-English code-mixing is often combined with Romanized Hindi, such as 'movie bahut achhi hai' or 'ye product bilkul worth it nahi hai'. The same sentiment can therefore be expressed using words that belong to different languages, different scripts, and informal spellings.

The SentiMix benchmark released through SemEval-2020 Task 9 provides a useful reproducible setting for this problem. Its Hindi-English and Spanish-English corpora contain sentence-level sentiment labels and word-level language information. The benchmark reports roughly 20,000 Hinglish examples and 19,000 Spanglish examples, with positive, negative, and neutral sentiment categories [4]. The original task attracted many participating systems and showed that BERT-like and ensemble approaches were among the strongest families.

1.3 Motivation

Three practical observations motivate this research. First, multilingual models must deal with language diversity rather than assuming that all input is clean English. Second, code-mixed text introduces tokenization and context problems that are not solved simply by adding more languages to a pre-trained model. Third, model accuracy alone is not sufficient for deployment: latency, memory, model size, and stability across languages also matter.

MuRIL was designed specifically for Indian languages and includes transliterated counterparts in its training strategy [5]. IndicBERT also demonstrates the value of language-specific multilingual pretraining for Indian languages [6]. These models provide strong foundations, but the research question remains: can their useful multilingual knowledge be compressed into a smaller model without losing the ability to interpret code-mixed sentiment?

1.4 Problem Statement

Existing sentiment-analysis systems face a trade-off between linguistic capability and computational efficiency. Large multilingual Transformers provide rich contextual representations but may be expensive for real-time or resource-constrained applications. Classical models are cheaper but may struggle with contextual meaning, code-switching, negation, transliteration, and informal social-media language. The problem addressed in this thesis is the design and evaluation of a compact Transformer-

based sentiment classifier that can handle multilingual and code-mixed social-media text while reducing computational requirements.

1.5 Research Questions

1. Can a compact Transformer retain competitive sentiment-classification performance when compared with larger multilingual Transformer models?
2. Does language-aware normalization improve sentiment classification on code-mixed and transliterated social-media text?
3. How much can model size and inference cost be reduced through knowledge distillation?
4. Does the compact model generalize across different code-mixed language pairs, particularly Hinglish and Spanglish?
5. What types of examples remain difficult for the proposed model, and can token-level explanations help identify those failure modes?

1.6 Research Objectives

6. To study the challenges of multilingual and code-mixed sentiment analysis in social-media text.
7. To design a lightweight Transformer architecture using a multilingual teacher-student framework.
8. To develop a preprocessing strategy that preserves sentiment-bearing informal and transliterated expressions.
9. To compare the proposed model with classical, recurrent, and full-size Transformer baselines.
10. To evaluate classification performance together with parameter count, latency, memory use, and robustness.
11. To investigate model explanations and common sources of classification error.
12. To develop a reproducible experimental protocol suitable for academic and engineering evaluation.

1.7 Scope

The primary experimental scope is sentence-level three-class sentiment classification: positive, negative, and neutral. The main benchmark is the SentiMix Hindi-English and Spanish-English data. The thesis does not attempt to solve every aspect of multilingual NLP, nor does it claim full understanding of sarcasm, humor, irony, or emotion beyond the available labels. Those phenomena are treated as sources of difficulty and error rather than as separately solved tasks.

1.8 Expected Contributions

- A compact Transformer-based architecture for multilingual and code-mixed sentiment classification.
- A reproducible language-aware preprocessing pipeline for noisy and transliterated social-media text.
- A teacher-student knowledge-distillation strategy designed to retain multilingual sentiment knowledge in a smaller model.
- An evaluation framework that treats accuracy and engineering efficiency as joint objectives.
- A systematic error and explainability analysis covering code-switching, negation, spelling variation, and ambiguous sentiment.

1.9 Organization of the Thesis

Chapter 1 introduces the problem. Chapter 2 reviews relevant research. Chapter 3 defines the research gap and formal objectives. Chapter 4 describes the benchmark data and preprocessing. Chapter 5 presents the proposed model. Chapter 6 defines the experimental design and evaluation metrics. Chapter 7 provides the result structure and analysis procedure. Chapter 8 focuses on explainability, robustness and errors. Chapter 9 discusses engineering deployment. Chapter 10 concludes the research and identifies future directions.

CHAPTER 2 – LITERATURE REVIEW

2.1 Evolution of Sentiment Analysis

Sentiment analysis has developed from manually constructed sentiment lexicons to statistical learning and then to deep contextual models. Lexicon-based methods are attractive because they are transparent and inexpensive, but they depend on predefined word-level polarity. A word may carry different sentiment depending on context, and a fixed dictionary has difficulty with slang, negation, sarcasm, and rapidly changing vocabulary.

Classical machine-learning approaches use representations such as Bag-of-Words, TF-IDF, character n-grams, or word n-grams. Support Vector Machines, logistic regression, and Naïve Bayes can perform well when the feature representation is appropriate. Their main limitation is that the features are usually less capable of representing long-range contextual relationships.

2.2 Deep Learning for Sentiment

Convolutional neural networks can learn local patterns such as sentiment-bearing phrases, while recurrent networks and LSTMs can model sequential information. These methods reduce the need for manual feature engineering, but they generally require task-specific training and may not match the transfer-learning advantages of large pre-trained Transformers.

2.3 Transformer Models

Transformers rely on self-attention rather than recurrence. This allows each token to interact with other tokens in the sequence, which is particularly useful when sentiment depends on a word that appears far away from the sentiment-bearing expression. BERT introduced deep bidirectional pretraining and showed the effectiveness of fine-tuning for downstream tasks [1].

XLM-R extended the multilingual Transformer idea by pretraining across many languages and showed strong cross-lingual transfer [7]. Such models provide a general multilingual foundation, but the inclusion of many languages also creates a capacity-allocation problem: a model must represent many linguistic systems while maintaining sufficient capacity for each one.

2.4 Indian-Language and Transliteration Models

MuRIL was developed specifically for Indian languages and incorporated translated and transliterated text pairs into its training strategy [5]. This is especially relevant to Romanized Indian-language social-media text. IndicBERT provides another important direction, using a multilingual ALBERT-style model trained on major Indian languages and designed to have fewer parameters than larger multilingual models [6]. These studies support the view that multilingual model design should reflect the actual linguistic environment of the target users.

2.5 Code-Mixed Sentiment Analysis

The SemEval-2020 SentiMix task established a widely used benchmark for sentiment analysis of Hindi-English and Spanish-English code-mixed tweets [4]. The task is important because it evaluates sentiment in the kind of noisy, mixed-language communication that conventional monolingual sentiment datasets often exclude.

Published systems from the SentiMix task illustrate that code-mixed sentiment can be addressed using feature engineering, CNNs, LSTMs, ensemble methods, and Transformers. For example, one Hinglish system reported that a classical NB-SVM approach could outperform a general-purpose Transformer in that particular benchmark setting, demonstrating that larger models are not automatically better on every code-mixed dataset [8]. This motivates a careful comparison rather than an assumption that Transformer models always dominate.

2.6 Knowledge Distillation

Knowledge distillation transfers information from a large teacher network to a smaller student. Instead of learning only from hard labels, the student can learn from the teacher's probability distribution over classes. This distribution contains information about class similarity and model uncertainty. DistilBERT showed the practical value of this approach for creating smaller Transformer models [2], while TinyBERT introduced a Transformer-specific distillation procedure that transfers intermediate representations as well as task outputs [3].

2.7 Research Gap from the Literature

Area	Existing Strength	Remaining Problem	Opportunity for This Thesis
Multilingual Transformers	Strong contextual transfer	High computation and capacity requirements	Compact student model
Code-mixed sentiment	Dedicated benchmarks exist	Noise, switching and transliteration remain difficult	Language-aware preprocessing
Distillation	Reduces model size	Performance may fall on unusual code-mixed input	Task-specific distillation
Indian-language NLP	MuRIL/IndicBERT improve coverage	Deployment cost remains relevant	Compact multilingual deployment
Evaluation	Accuracy/F1 widely reported	Efficiency often underreported	Joint accuracy-efficiency evaluation
Interpretability	Attention and post-hoc tools available	Errors remain difficult to diagnose	Token-level explanation and error analysis

2.8 Summary

The literature shows clear progress from lexical methods to multilingual Transformers. It also shows that model scale alone does not guarantee robust code-mixed sentiment analysis. The present study therefore focuses on a practical middle ground: preserve the contextual strength of multilingual Transformers while reducing their computational burden.

CHAPTER 3 – RESEARCH GAP, PROBLEM DEFINITION AND OBJECTIVES

3.1 Consolidated Research Gap

The first research gap is the mismatch between benchmark-oriented model development and deployment constraints. Many studies emphasize classification performance but do not report model size, memory use, or latency in sufficient detail. A model that improves F1 by a small margin but requires several times the memory may not be the best engineering solution.

The second gap is the difficulty of code-mixed and transliterated text. Romanized Hindi does not follow a single spelling standard. The same word may appear in several forms, and users may insert English words, abbreviations, emojis, and punctuation into the same utterance. A preprocessing strategy that aggressively cleans text can accidentally remove useful sentiment information.

The third gap is cross-language robustness. A model may perform well on Hinglish but behave differently on Spanglish or another code-mixed pair. A multilingual framework should therefore be evaluated across more than one language pair where the benchmark permits it.

3.2 Formal Problem Definition

Let $D = \{(x_i, y_i)\}$ for $i = 1, \dots, N$ represent a dataset of social-media texts x_i and sentiment labels $y_i \in \{\text{positive, negative, neutral}\}$. Each x_i may contain tokens from multiple languages, transliteration, emojis, punctuation, abbreviations, or misspellings. The objective is to learn a classifier $f_\theta(x)$ that predicts y while minimizing both classification loss and deployment cost.

The proposed optimization objective is expressed conceptually as: $L_{\text{total}} = \alpha L_{\text{label}} + \beta L_{\text{distill}} + \gamma L_{\text{hidden}} + \delta L_{\text{language}}$, where L_{label} is the ordinary cross-entropy loss against the gold sentiment label, L_{distill} is the divergence between teacher and student class distributions, L_{hidden} represents optional intermediate-representation alignment, and L_{language} encourages consistent representations across the language conditions represented in the training set. α , β , γ and δ are tunable weights.

For deployment, the study also considers an efficiency objective. Let P denote the number of trainable parameters, T the average inference time per example, and M the peak inference memory. A practical model should improve sentiment performance without allowing these quantities to grow unnecessarily.

3.3 Research Hypotheses

13. H1: The proposed lightweight Transformer will achieve a macro-F1 score that is competitive with a larger multilingual Transformer baseline.
14. H2: Knowledge distillation will reduce model size and inference cost compared with the teacher model.
15. H3: Language-aware normalization will improve performance on noisy and transliterated code-mixed text compared with aggressive generic cleaning.
16. H4: The proposed model will maintain useful performance across both Hinglish and Spanglish rather than overfitting to a single language pair.
17. H5: Error analysis will show that residual failures are concentrated around sarcasm, implicit sentiment, ambiguous code-switching, and highly noisy text.

3.4 Novelty Positioning

The novelty of this work is not the invention of the Transformer itself. Instead, the contribution lies in combining multilingual representation, code-mixed-aware preprocessing, task-specific knowledge distillation, and deployment-oriented evaluation into one reproducible framework. The research is therefore positioned as an applied AI engineering contribution.

CHAPTER 4 – DATASET AND DATA PREPARATION

4.1 Benchmark Selection

The primary benchmark should be the SentiMix dataset from SemEval-2020 Task 9. The benchmark contains Hindi-English and Spanish-English code-mixed social-media text with positive, negative, and neutral sentiment labels and word-level language annotations [4]. The published overview reports approximately 20K Hinglish examples and 19K Spanglish examples.

The dataset should be obtained from the official or authorized distribution associated with the SemEval task. The researcher must retain the original train/development/test separation when the benchmark provides it and must not use test labels during training or model selection.

4.2 Data Audit

18. Count the number of records in each language pair.
19. Count positive, negative, and neutral examples.
20. Measure average and maximum token length.
21. Measure the proportion of examples containing Romanized Indic words, emojis, URLs, hashtags, usernames, repeated characters, and unusual punctuation.
22. Check duplicate records and near-duplicates without changing the official test set.
23. Record missing or malformed entries before preprocessing.

4.3 Language-Aware Normalization

The proposed preprocessing pipeline avoids aggressive normalization. URLs and usernames may be replaced by stable placeholders such as <URL> and <USER>. Repeated whitespace is normalized. Repeated punctuation is retained when it may carry sentiment, while extreme repetition can be capped. Emojis are preserved or converted into textual descriptions because they can provide sentiment information.

Romanized words should not simply be translated into English using a generic dictionary. Transliteration is highly variable, and an incorrect translation can change the sentiment. Instead, the text is passed through the multilingual tokenizer after lightweight normalization. Where a language identifier is available, it can be used as an auxiliary feature for analysis rather than as a hard rule that forces every token into one language.

4.4 Data Splitting

When the benchmark already defines train, development, and test partitions, those partitions should be preserved. If a separate internal validation split is required, it should be created only from the training portion. Stratification by sentiment label is recommended. All preprocessing rules must be fitted or selected using training data only.

4.5 Class Imbalance

Class distribution must be reported before training. If the classes are imbalanced, the study may use class-weighted cross-entropy or controlled sampling. Oversampling should be applied only to the training data. Macro-F1 should remain a primary metric because it gives equal importance to each sentiment class.

4.6 Data Privacy and Ethics

Social-media datasets can contain personal information even when usernames are removed. The study should use the benchmark under its stated research conditions, avoid attempting to identify individuals, and avoid reproducing unnecessary user-generated content in the thesis. Examples in the error-analysis chapter should be anonymized or paraphrased where appropriate.

CHAPTER 5 – PROPOSED LIGHTWEIGHT TRANSFORMER FRAMEWORK

5.1 Overview

The proposed system, referred to in this thesis as LightMuRIL, is a compact student Transformer trained with supervision from a stronger multilingual teacher. MuRIL is selected as the principal teacher because it was designed for Indian languages and explicitly incorporated transliterated text during pretraining [5]. For the Spanglish evaluation, the teacher can be supplemented or replaced by XLM-R where the researcher wants a broader multilingual baseline [7].

The student is intentionally smaller. A practical configuration for the experimental study is a six-layer Transformer with reduced hidden dimension and shared multilingual vocabulary. The exact configuration should be reported in the final paper after the implementation is fixed. The important design principle is that the student must be materially smaller than the teacher and must be evaluated under the same task conditions.

5.2 System Architecture

The proposed pipeline is:

Raw social-media text $\rightarrow$ lightweight normalization $\rightarrow$ tokenizer $\rightarrow$ compact Transformer encoder $\rightarrow$ attention pooling / [CLS] representation $\rightarrow$ dropout $\rightarrow$ three-class sentiment head $\rightarrow$ sentiment probabilities + confidence + explanation.

During training, a teacher model processes the same input. The student receives three sources of learning: the gold sentiment label, the teacher's softened class distribution, and optionally an intermediate representation from the teacher. This makes the student learn both the final task and some of the teacher's internal decision structure.

5.3 Teacher–Student Distillation

Let z_t and z_s denote the teacher and student logits. With temperature τ , the softened probabilities are $p_t = \text{softmax}(z_t/\tau)$ and $p_s = \text{softmax}(z_s/\tau)$. The distillation loss can be written as $\text{KL}(p_t \parallel p_s)$, multiplied by τ^2 to preserve useful gradient scale.

The combined objective is: $L = \alpha L_{\text{CE}}(y, p_s) + \beta \tau^2 \text{KL}(p_t \parallel p_s) + \gamma L_{\text{rep}}$. Here L_{CE} is the gold-label cross-entropy loss, the second term aligns student and teacher predictions, and L_{rep} is an optional representation-alignment term. The coefficients should be selected using the development set rather than the test set.

5.4 Language-Aware Preprocessing Module

- Preserve sentiment-bearing punctuation and emojis.
- Replace URLs and user mentions with stable placeholders.
- Normalize repeated whitespace without deleting meaningful tokens.
- Normalize extreme character repetition only after measuring its effect.
- Retain Romanized Indic words rather than translating them blindly.
- Use tokenizer behavior as an evaluation signal by reporting unknown/subword fragmentation statistics.
- Record language-pair metadata for per-language analysis.

5.5 Sentiment Classification Head

The final hidden representation is passed through dropout and a small feed-forward classification layer. The output dimension is three. Softmax converts logits into probabilities. The predicted class is the class with the highest probability. Confidence should be reported separately from accuracy because a high-confidence error is more concerning than an uncertain prediction.

5.6 Optional Calibration

If resources permit, temperature scaling can be applied to the development set after training. Calibration should not change the predicted class in most cases, but it can make confidence values more meaningful. Expected Calibration Error (ECE) can be reported as an additional engineering metric.

5.7 Algorithm

24. Load the multilingual teacher and tokenizer.
25. Load and audit the code-mixed training data.

26. Apply language-aware normalization.
27. Tokenize the same normalized input for teacher and student.
28. Generate teacher logits without updating teacher parameters.
29. Generate student logits and hidden representations.
30. Compute gold-label loss.
31. Compute teacher-student distillation loss.
32. Optionally compute intermediate representation loss.
33. Backpropagate only through the student.
34. Evaluate on the development set after each epoch.
35. Select the best checkpoint using macro-F1 with efficiency as a secondary criterion.
36. Run the final model once on the untouched test set.
37. Measure accuracy, macro-F1, class-wise F1, latency, memory, and parameter count.
38. Perform error and explanation analysis.

5.8 Why the Framework is Lightweight

The term lightweight is operational rather than rhetorical. The final manuscript should report the exact number of parameters, model file size, average inference latency on a defined hardware platform, and peak memory. A model should not be called lightweight merely because it is smaller than an arbitrary reference. The study therefore defines efficiency through measurable quantities.

CHAPTER 6 – EXPERIMENTAL DESIGN AND EVALUATION

6.1 Experimental Baselines

Model	Role	Expected Strength	Expected Limitation
TF-IDF + Logistic Regression	Classical baseline	Very low cost	Limited contextual understanding
TF-IDF + Linear SVM	Strong classical baseline	Good sparse-text performance	Feature-dependent
BiLSTM	Neural sequence baseline	Sequential context	Training/inference cost
XLM-R / multilingual Transformer	Large Transformer baseline	Broad multilingual transfer	Higher resource use
MuRIL fine-tuned	Indian-language teacher	Strong transliteration support	Large model
LightMuRIL	Proposed model	Efficiency + multilingual context	Potential compression loss

6.2 Hardware and Software

The final experimental report must state the exact hardware used, including CPU model, GPU model if applicable, RAM, operating system, Python version, PyTorch version, Transformers version, and CUDA version where relevant. This is essential because latency and memory results are hardware-dependent.

6.3 Suggested Hyperparameters

Parameter	Recommended Starting Value	Selection Method
Maximum sequence length	128	Development-set validation

Batch size	16–32	Hardware constrained
Learning rate	2e-5 to 5e-5	Development-set search
Epochs	3–8	Early stopping
Dropout	0.1–0.3	Development-set search
Distillation temperature	2–5	Development-set search
Teacher loss weight	0.4–0.8	Development-set search
Random seeds	At least 3	Stability analysis

These are starting points, not claimed optimal values. The final thesis must report the values actually used. Hyperparameter search should be modest and transparent so that the study remains reproducible.

6.4 Evaluation Metrics

Accuracy is the fraction of correctly classified examples. Precision measures the proportion of predicted examples of a class that are correct. Recall measures the proportion of true examples of a class that are recovered. F1 combines precision and recall through their harmonic mean. Because sentiment datasets can be imbalanced, macro-F1 is emphasized.

For engineering evaluation, parameter count, model size, peak inference memory, average latency, and throughput should also be measured. Latency should be reported over multiple runs after a warm-up period and should specify batch size and sequence length.

6.5 Statistical Reliability

A single random seed can produce an unstable estimate. The recommended protocol is to train each key model with at least three seeds. Report mean and standard deviation for macro-F1 and accuracy. If computational resources allow, bootstrap confidence intervals can be added.

6.6 Ablation Study

The ablation study is essential because it separates the contribution of each component. The following variants are recommended:

- Student without distillation.
- Student with distillation but without language-aware normalization.
- Student with normalization but without representation alignment.
- Full LightMuRIL model.
- Full model without emoji preservation.
- Full model with a smaller hidden size.

6.7 Cross-Language Evaluation

The same training protocol should be evaluated separately on Hinglish and Spanglish. The final analysis should not simply average the two scores. Per-language results reveal whether the model is genuinely multilingual or whether its performance is dominated by the language pair with the stronger representation.

IMPORTANT: The numerical results in this draft are PROVISIONAL EXPECTED VALUES, not measured experimental results. They are provided because experiment logs/models were not available. Verify or replace them with actual runs before submission.

CHAPTER 7 – RESULTS AND ANALYSIS

7.1 Important Integrity Statement

Because the original experiment logs were unavailable, the result tables below use provisional expected values to complete the thesis draft. They are intended for planning/draft presentation only and must be replaced with actual measured results before journal submission.

7.2 Main Classification Results

Model	Language	Accuracy	Macro-F1	Weighted-F1	Params (M)	Latency (ms)	Memory (MB)
TF-IDF + Logistic Regression	Hinglish	67.8%	64.9%	66.3%	0.01	0.4	18
TF-IDF + Linear SVM	Hinglish	69.5%	67.1%	68.4%	0.02	0.5	20
BiLSTM	Hinglish	68.9%	66.5%	67.8%	3.2	4.8	72
XLM-R / large multilingual	Hinglish	73.1%	71.0%	72.0%	278	9.8	980
MuRIL	Hinglish	72.6%	70.5%	71.6%	236	8.4	860
LightMuRIL	Hinglish	71.8%	69.7%	70.9%	18.6	3.1	210
LightMuRIL	Spanglish	74.0%	72.1%	73.2%	18.6	3.2	215

The final analysis should answer three questions. First, whether the proposed model reaches competitive classification performance. Second, whether the reduction in computational cost is meaningful. Third, whether the trade-off is acceptable for a practical deployment.

7.3 Per-Class Results

Model	Language	Positive F1	Negative F1	Neutral F1	Macro-F1
MuRIL	Hinglish	72.8%	68.4%	70.3%	70.5%
LightMuRIL	Hinglish	72.1%	67.6%	69.4%	69.7%
MuRIL	Spanglish	75.0%	69.2%	72.0%	72.1%
LightMuRIL	Spanglish	75.8%	70.0%	70.5%	72.1%

Neutral sentiment is expected to be particularly informative because neutral examples often contain fewer strongly polar words. The researcher should therefore discuss whether compression affects the model's ability to distinguish neutral from weakly positive or weakly negative text.

7.4 Confusion Matrix Analysis

For each language pair, a 3×3 confusion matrix should be included. The discussion should focus on the largest off-diagonal errors. For example, if negative examples are frequently classified as neutral, the researcher should inspect whether negation, implicit sentiment, or insufficient context is responsible.

7.5 Efficiency Results

Model	Parameters	Model Size	Peak Memory	Avg. Latency	Throughput
Large	278M	1,060 MB	9.8 ms	1,020 ex/s	provisional

multilingual Transformer					expected value
MuRIL	236M	860 MB	8.4 ms	1,190 ex/s	provisional expected value
LightMuRIL	18.6M	210 MB	3.1 ms	3,230 ex/s	provisional expected value

The key efficiency result should be expressed as a relative reduction rather than a vague statement. For example, parameter reduction can be calculated as $100 \times (P_{\text{teacher}} - P_{\text{student}}) / P_{\text{teacher}}$. The same approach can be used for memory and latency. The final paper should state the hardware and measurement procedure next to every efficiency claim.

7.6 Ablation Results

Variant	Macro-F1	Change vs Full Model	Interpretation
Student without distillation	65.8%	-3.9 pp	Measures value of teacher knowledge
Student + distillation	68.7%	-1.0 pp	Core compression effect
Without language-aware normalization	67.2%	-2.5 pp	Measures preprocessing contribution
Without representation alignment	68.1%	-1.6 pp	Measures intermediate-feature contribution
Full LightMuRIL	69.7%	Reference	Complete proposed framework

7.7 Expected Interpretation Framework

If the proposed model produces slightly lower F1 than the teacher but substantially lower latency and memory, the correct conclusion is not that the model is universally better. The stronger claim is that the model provides a useful accuracy-efficiency trade-off. If it matches or exceeds the teacher, the result is stronger and should be investigated for possible benefits from regularization or preprocessing. If it performs substantially worse, the thesis should report that outcome honestly and identify the likely causes.

CHAPTER 8 – EXPLAINABILITY, ROBUSTNESS AND ERROR ANALYSIS

8.1 Why Explainability Matters

A sentiment classifier may be accurate overall but still make systematic mistakes. For research and deployment, it is useful to know which tokens or phrases influenced a decision. Explanation methods can be used to highlight influential tokens, but the explanations should not be treated as proof of causality. They are diagnostic tools.

8.2 Token-Level Analysis

The study should select representative correct and incorrect examples. For each example, display the predicted class, confidence, true class, language pair, and the most influential tokens. The analysis should examine whether the model focuses on meaningful sentiment-bearing words or on irrelevant artifacts such as punctuation, usernames, or frequent platform-specific terms.

8.3 Error Categories

Error Type	Typical Cause	Diagnostic Question
Sarcasm / irony	Literal words conflict with intended sentiment	Does the model rely too heavily on lexical polarity?
Negation	Polarity changes because of 'not', 'nahi', etc.	Does the model connect negation to the sentiment phrase?
Code switching	Sentiment is distributed across languages	Does the model integrate both language segments?
Transliteration	Non-standard Romanized spelling	Does tokenization preserve useful subwords?
Emoji dependence	Emotion is expressed visually	Does removing emojis change the decision?
Implicit sentiment	No obvious sentiment word	Does the model use contextual information?
Ambiguous context	Meaning depends on missing conversation	Is the input itself insufficient?

8.4 Robustness Tests

39. Remove emojis and measure the change.
40. Normalize repeated punctuation and compare predictions.
41. Introduce mild spelling variation into a controlled sample.
42. Replace common Romanized forms with alternative spellings.
43. Remove URLs and usernames.
44. Evaluate shorter and longer sequences separately.
45. Compare confidence distributions for clean and noisy inputs.

The goal is not to make the model immune to every perturbation. The goal is to determine which forms of noise cause meaningful degradation and whether the proposed preprocessing and distillation strategy makes the model more stable.

8.5 Bias and Fairness Considerations

Sentiment datasets can contain topic, demographic, cultural, or platform-specific biases. A model may learn associations that reflect the dataset rather than genuine sentiment. The study should therefore avoid claims of universal fairness. Where metadata permits, performance should be inspected across language pairs and major topic groups. Sensitive demographic inference should not be introduced merely for the purpose of evaluation.

CHAPTER 9 – ENGINEERING DEPLOYMENT AND PRACTICAL DISCUSSION

9.1 Deployment Motivation

Sentiment analysis is often used as a component of larger information systems: customer-feedback dashboards, communication monitoring, help-desk prioritization, product-review analysis, and multilingual social-media analytics. In such systems, a model may process thousands of short messages. Small reductions in per-message latency and memory can therefore become important at scale.

9.2 Edge and Resource-Constrained Use

A compact Transformer can be deployed on a CPU or a modest GPU more easily than a large multilingual model. The final deployment claim should be based on measured latency rather than theoretical parameter count alone. Quantization may be investigated as a future optimization, but it should not be included as a contribution unless it is experimentally evaluated.

9.3 API or Streaming Architecture

A practical architecture can expose the classifier as a service. Incoming text is normalized, tokenized, classified, and returned with sentiment probabilities. The system can optionally return an explanation for human review. For high-volume applications, batching can improve throughput, while for interactive applications single-example latency is more important.

9.4 Accuracy–Efficiency Trade-off

The central engineering question is not simply which model has the highest F1. A useful decision can be framed as a Pareto comparison: models that are both less accurate and more expensive are dominated, while models that sacrifice a small amount of accuracy for a large reduction in computation may be attractive. This framing makes the research relevant to practical AI deployment.

9.5 Limitations

- The benchmark is based on social-media text and may not represent every real-world domain.
- Three sentiment classes do not capture the full range of human emotion.
- Sarcasm and irony remain difficult without conversational context.
- Knowledge distillation can lose information contained in the teacher.
- Latency depends on hardware, software versions, batch size, and sequence length.
- Cross-language generalization cannot be guaranteed beyond the evaluated language pairs.
- Dataset bias may remain even when overall F1 is high.

CHAPTER 10 – CONCLUSION AND FUTURE WORK

10.1 Conclusion

This thesis proposes a practical approach to multilingual and code-mixed sentiment analysis based on a lightweight Transformer. The central idea is straightforward: large multilingual models contain useful contextual knowledge, but deployment does not always permit their full computational cost. Knowledge distillation provides a mechanism for transferring useful behavior into a smaller student model.

The research places particular emphasis on code-mixed social-media language because this setting exposes weaknesses that are less visible in clean monolingual datasets. Hindi-English and Spanish-English data provide a reproducible basis for studying language switching, transliteration, informal spelling, and sentiment ambiguity. The proposed framework combines language-aware preprocessing, multilingual representation, task-specific distillation, and deployment-oriented evaluation.

The most important methodological principle is that the final conclusion must be driven by measured results. If the compact model achieves competitive F1 with substantially lower computation, it supports

the proposed accuracy-efficiency trade-off. If the model loses too much performance, the research should document that limitation rather than hiding it. Either outcome can provide useful engineering evidence when the experiment is carefully controlled and transparently reported.

10.2 Future Work

- Extend the framework to additional Indic-English code-mixed language pairs.
- Investigate quantization and pruning for further deployment efficiency.
- Add conversational context for sarcasm and irony detection.
- Study continual learning for rapidly changing social-media vocabulary.
- Explore multimodal sentiment using text, emoji, image, and short-video signals.
- Develop privacy-preserving or federated training for distributed social-media analytics.
- Investigate uncertainty estimation for human-in-the-loop decision systems.
- Create a larger, carefully balanced multilingual benchmark with transparent annotation guidelines.

10.3 Final Research Contribution

The intended contribution of the study is a reproducible engineering framework rather than a claim that one model solves multilingual sentiment analysis. The work connects modern Transformer NLP with practical AI deployment requirements and provides a structured way to measure the trade-off between linguistic capability and computational cost.

REFERENCES

1. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
2. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv:1910.01108. <https://doi.org/10.48550/arXiv.1910.01108>
3. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for natural language understanding. Findings of EMNLP 2020, 4163–4174. <https://doi.org/10.18653/v1/2020.findings-emnlp.372>
4. Patwa, P., Aguilar, G., Kar, S., Pandey, S., PYKL, S., Gambäck, B., Chakraborty, T., Solorio, T., & Das, A. (2020). SemEval-2020 Task 9: Overview of sentiment analysis of code-mixed tweets. Proceedings of SemEval 2020. <https://aclanthology.org/2020.semeval-1.100/>
5. Khanuja, S., Bansal, D., Mehtani, S., Khosla, S., Dey, A., Gopalan, B., Margam, D. K., Aggarwal, P., Nagipogu, R. T., Dave, S., Gupta, S., Gali, S. C. B., Subramanian, V., & Talukdar, P. (2021). MuRIL: Multilingual representations for Indian languages. arXiv:2103.10730. <https://doi.org/10.48550/arXiv.2103.10730>
6. Kakwani, D., Kunchukuttan, A., Golla, S., N.C., G., Bhattacharyya, A., Khapra, M. M., & Kumar, P. (2020). IndicNLP Suite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. Findings of EMNLP 2020.
7. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. Proceedings of ACL 2020, 8440–8451. <https://doi.org/10.18653/v1/2020.acl-main.747>
8. Bhange, M., & Kasliwal, N. (2020). HinglishNLP at SemEval-2020 Task 9: Fine-tuned language models for Hinglish sentiment detection. Proceedings of SemEval 2020.

9. Advani, L., Lu, C., & Maharjan, S. (2020). C1 at SemEval-2020 Task 9: Sentiment analysis for code-mixed social media text using feature engineering. Proceedings of SemEval 2020.
10. Srivastava, V., & Singh, M. (2020). IIT Gandhinagar at SemEval-2020 Task 9: Code-mixed sentiment classification using candidate sentence generation and selection. Proceedings of SemEval 2020.
11. Aparaschivei, L., Palihovici, A., & Gîfu, D. (2020). FII-UAIC at SemEval-2020 Task 9: Sentiment analysis for code-mixed social media text using CNN. Proceedings of SemEval 2020.
12. Doddapaneni, S., Aralikatte, R., Ramesh, G., Goyal, S., Khapra, M. M., Kunchukuttan, A., & Kumar, P. (2023). Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages. Proceedings of ACL 2023, 12402–12426. <https://doi.org/10.18653/v1/2023.acl-long.693>
13. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., & Wang, W. (2022). Language-agnostic BERT sentence embedding. Proceedings of ACL 2022. <https://aclanthology.org/2022.acl-long.62/>
14. International Prime Publications. (2026). Artificial Intelligence and Electrical & Electronics Engineering: AIEEE Open Access – Instructions for Authors. ISSN 3069-0706. <https://www.primeopenaccess.com/international-journals/artificial-intelligence-and-electrical-electronics-engineering-aieee-open-access-instructions.asp>