B.TECH FINAL PROJECT REPORT

ROBUST SENTIMENT ANALYSIS FOR MULTILINGUAL AND CODE-SWITCHED TEXT

A Comparative Study: Traditional ML vs. Deep Learning

SUBMITTED BY

Vinay sharma

MIS No: 112215199

GUIDED BY

Prof. Anupriya

Department of CSE, IIIT Pune

December 2025

PRESENTATION OUTLINE

- **01.** Introduction & Motivation
- **02.** Problem Statement
- **03.** Dataset Overview & EDA
- **04.** Methodology (Baseline vs Proposed)
- **05.** System Architecture
- **06.** Experimental Results
- **07.** Conclusion & Future Scope

INTRODUCTION

The Modern Challenge

Sentiment analysis is no longer just about English text. Modern social media data is **multilingual** and informal. Users frequently mix languages (Hindi + English) in the same sentence.

The Scale

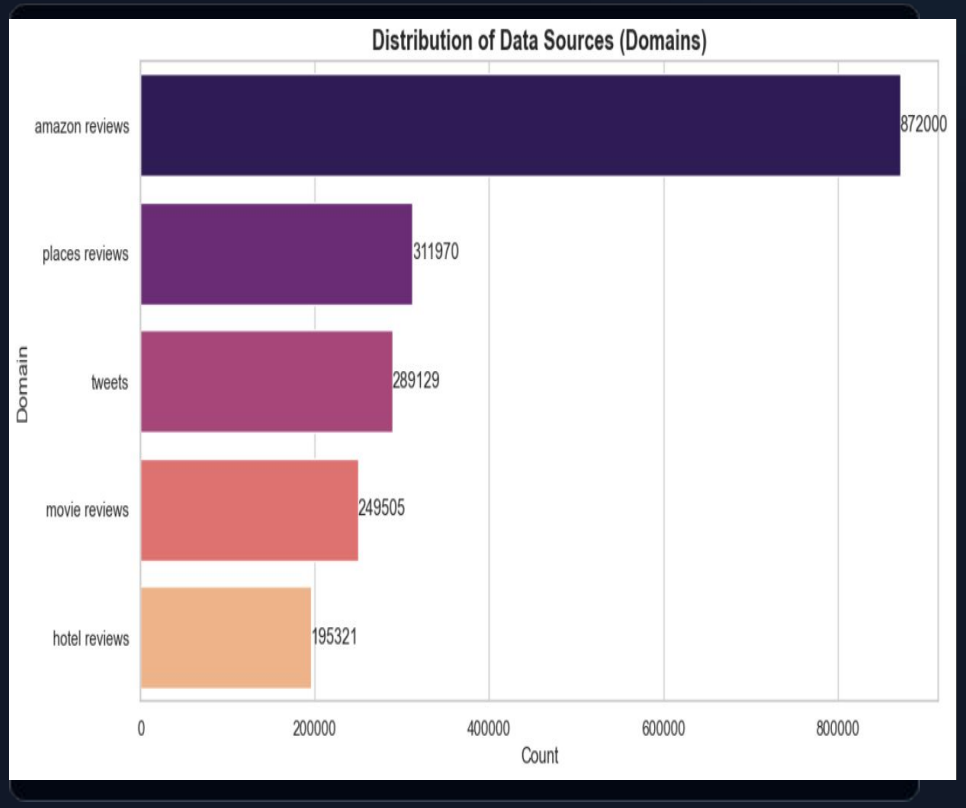
This project tackles a massive dataset of **3.1 Million** user comments, necessitating robust computational methods.

Goal: Develop a system that understands context in "messy" real-world text where traditional dictionaries fail.

MOTIVATION

Why this study?

- **Code-Switching:** Indian users often say *"Product bahut good hai"*. Traditional models see "bahut" as noise.
- **False Friends:** Words like "Gift" mean "Present" in English but "Poison" in German. Without context, classification fails.
- **Business Need:** Companies need accurate insights from the 75% of users who don't post in pure standard English.



PROBLEM STATEMENT

The Core Issue

Accurate classification of sentiment in a 3.1 million row dataset characterized by **high linguistic noise** and mixed scripts.

Limitations of Current Tools

Off-the-shelf tools rely on translation, which loses nuance (slang, sarcasm, local context).

```
Input: "Movie was badiya but songs bakwas"
Standard Model: Neutral (Unknown words)
Desired Output: Mixed/Negative
```

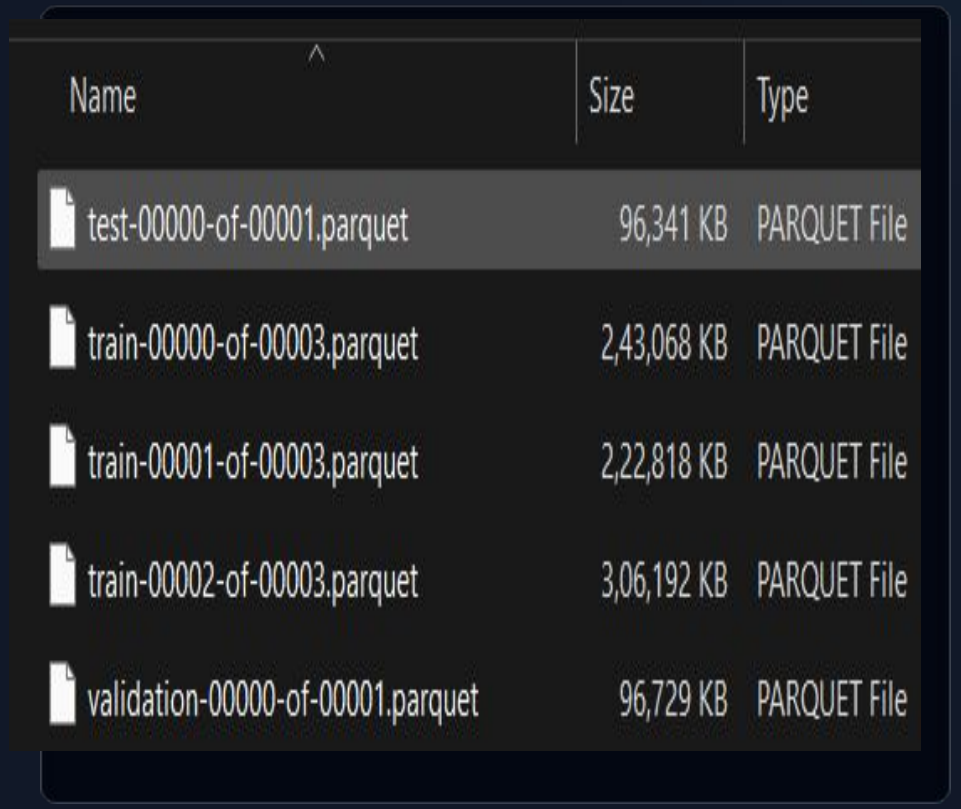
RESEARCH OBJECTIVES

- **Establish a Baseline:** Implement TF-IDF + Naive Bayes to quantify the difficulty of the dataset.
- **Implement Deep Learning:** Fine-tune **XLM-RoBERTa** (pre-trained on 100 languages) to handle context.
- **Optimize Training:** Use **Stratified Sampling** to balance Positive, Negative, and Neutral classes (200k samples).
- **Compare:** Rigorously evaluate accuracy improvements on a held-out test set.

DATASET OVERVIEW

We utilized a large-scale unstructured dataset.

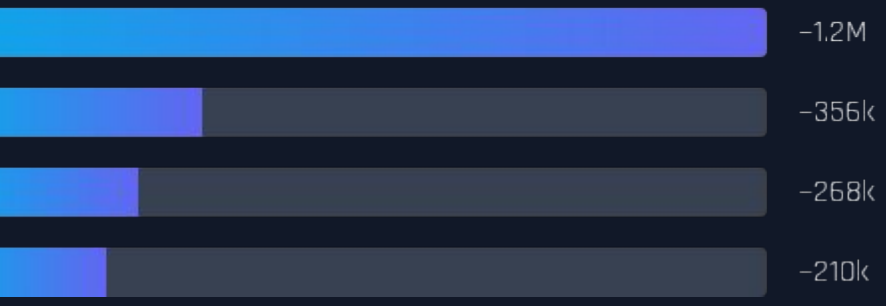
Property	Details
Total Records	3,147,478 (3.1 Million)
Format	CSV / Parquet
Key Columns	text, label, source, language
Languages	English, Hindi, French, Mixed

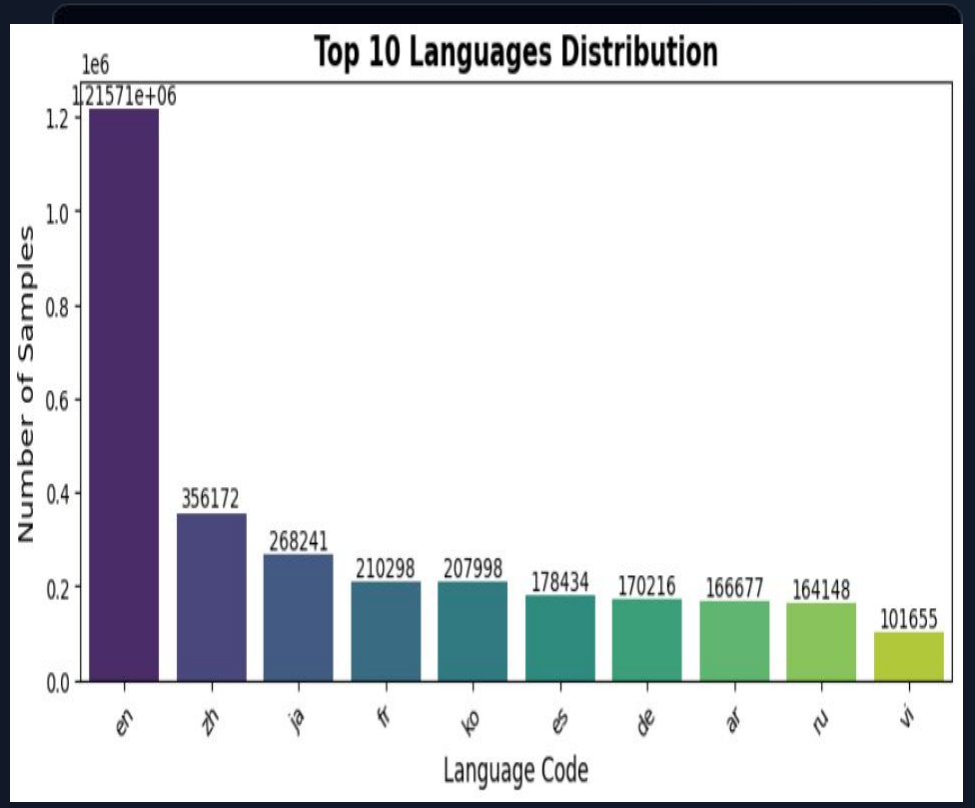


EDA: LANGUAGE DISTRIBUTION

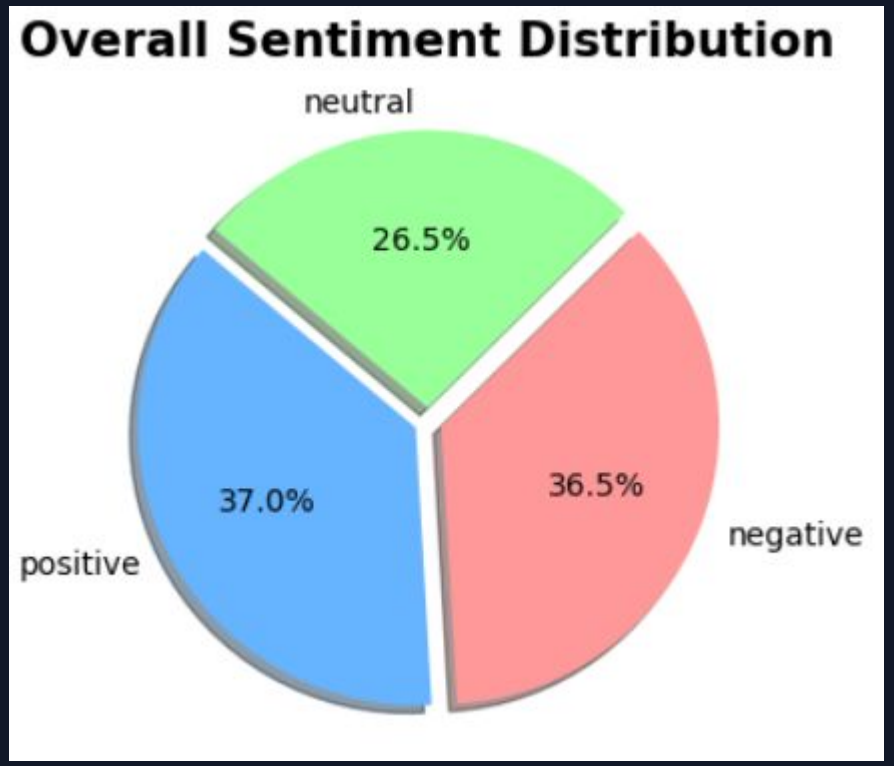
Language Breakdown

Based on Figure 3.3.2 in the report, the dataset is dominated by English but contains significant non-English data.





EDA: SENTIMENT DISTRIBUTION



Class Distribution (Fig 3.3.3)

The dataset is relatively balanced, but "Neutral" is the slight majority.

- **Neutral:** 26.5%
- **Positive:** 37%
- **Negative:** 36.5%

*Stratified sampling was used to handle this during training.

PREPROCESSING PIPELINES

Pipeline A (Baseline)

Aggressive Cleaning: Removed all non-English characters, URLs, and stop-words.

Result: Loss of context (e.g., "not" might be removed as a stop-word).

Pipeline B (Proposed)

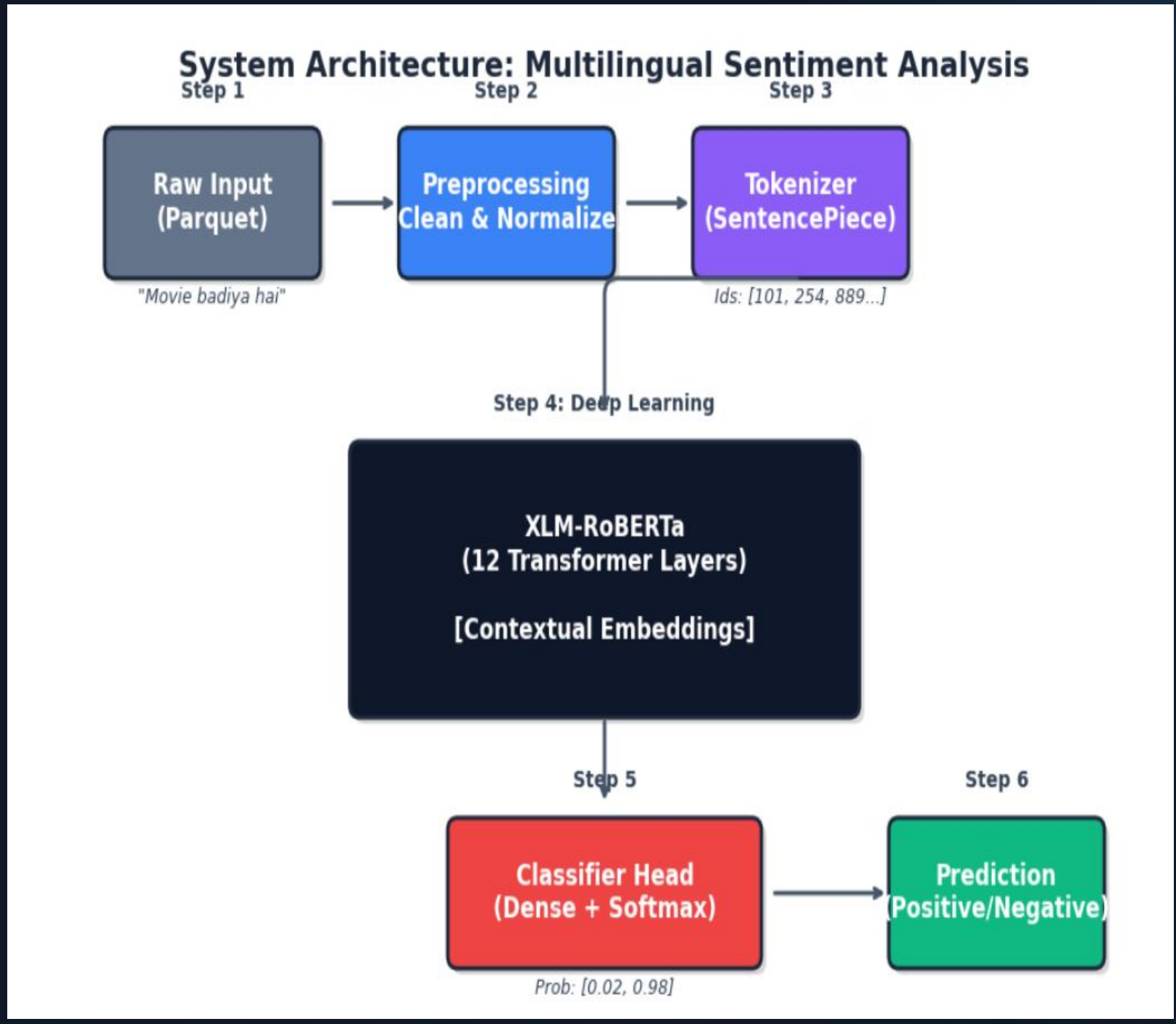
Minimal Cleaning: Kept Hindi characters and punctuation.

Reason: XLM-R Tokenizer handles raw text and uses punctuation for context.

SYSTEM ARCHITECTURE

A sequential flow from raw data to prediction.

- **Ingestion:** Load Parquet Files.
- **Tokenizer:** SentencePiece (250k vocab).
- **Model:** XLM-RoBERTa (12 Layers).
- **Classifier:** Dense Layer + Softmax.
- **Output:** Sentiment Label.



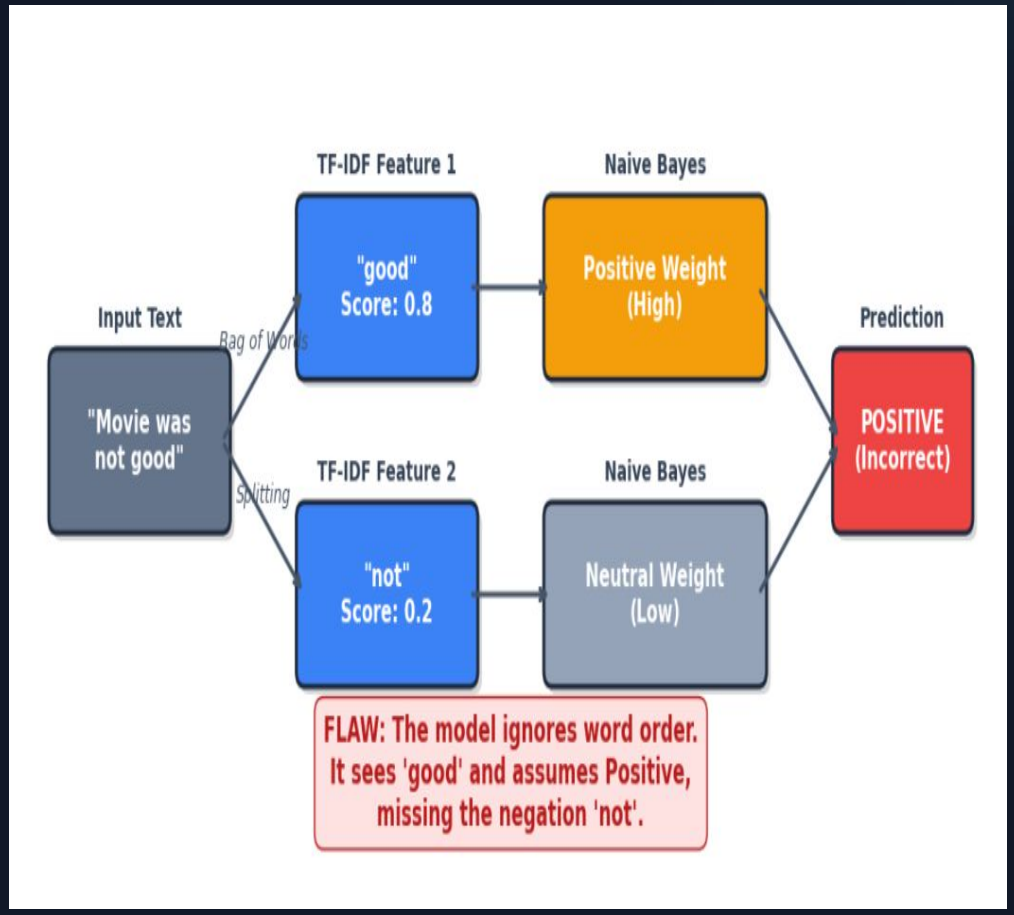
METHODOLOGY: THE BASELINE

TF-IDF + Naive Bayes

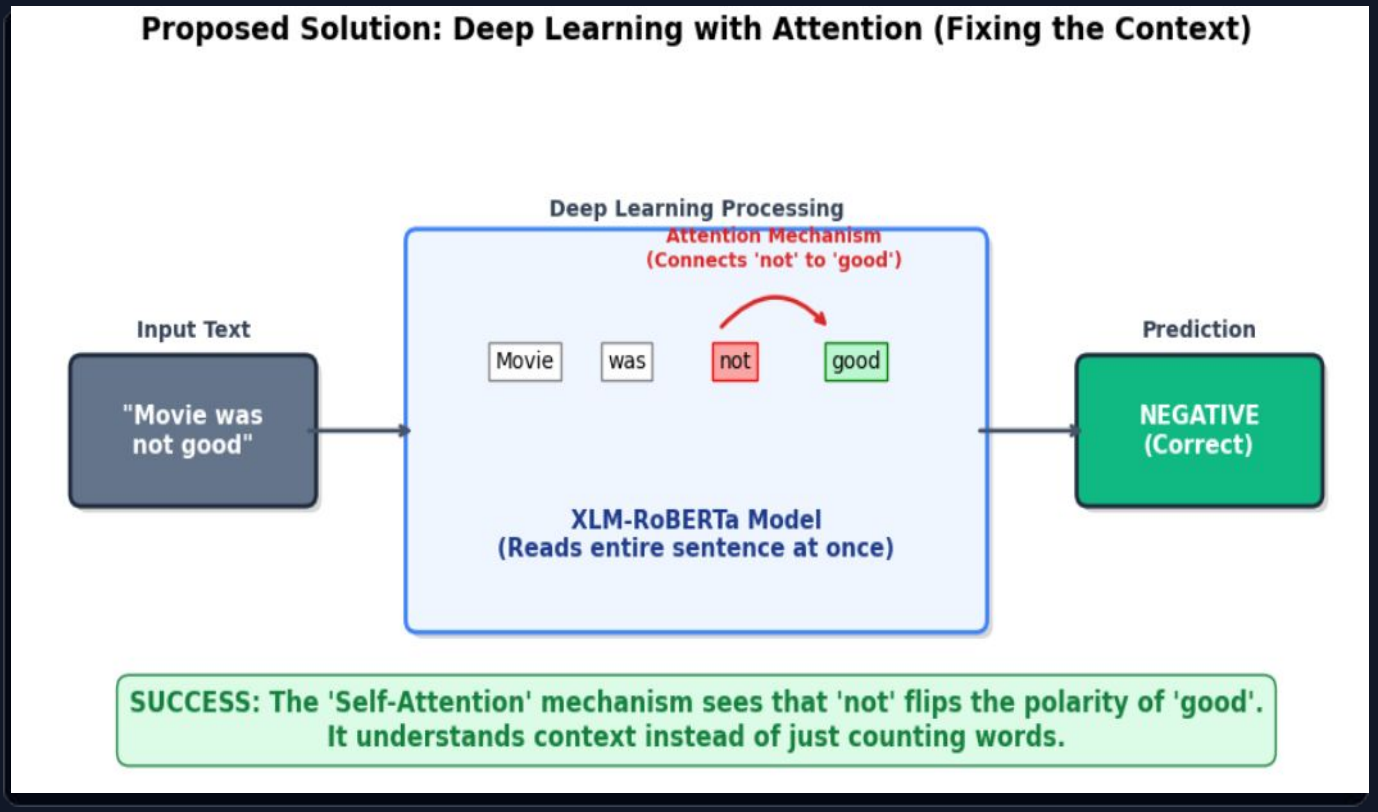
We started with a traditional statistical approach.

- **TF-IDF:** Converts text to numbers based on word frequency.
- **Naive Bayes:** Calculates probability assuming words are independent.

Flaw: It sees "good" and "not" as separate. It doesn't understand that "not good" flips the meaning.



METHODOLOGY: PROPOSED SOLUTION



Fine-Tuning XLM-RoBERTa

XLM-R is a Transformer model pre-trained on 2.5TB of data across **100 languages**.

- **Contextual:** Reads the whole sentence at once.
- **Cross-Lingual:** Applies knowledge from English to Hindi.
- **Tokenizer:** Uses SentencePiece to handle unknown words.

TRAINING STRATEGY

Stratified Sampling

We created a balanced training set of **200,000 samples** (~66k per class) to avoid bias towards the Neutral class.

Optimization

Optimizer: AdamW

Learning Rate: 2e-5

Scheduler: Linear Decay

Batching

Batch Size: 8

Gradient Accumulation: 4

(Effective Batch Size: 32)

Component	Specification
Platform	Local Workstation / Kaggle Kernels
GPU	NVIDIA GeForce RTX 4060 / Tesla T4 (x2)
VRAM	8 GB (Local) / 16 GB (Cloud)
RAM	16 GB
Language	Python 3.10
Key Libraries	PyTorch, Transformers (Hugging Face), Scikit-learn, Pandas

HARDWARE SPECIFICATIONS

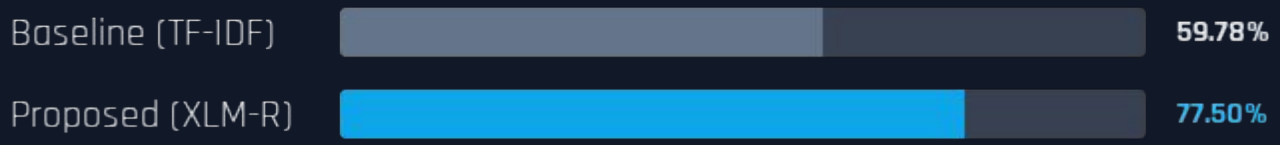
Transformer training is computationally intensive. We utilized high-performance GPUs.

Component	Spec
GPU (Local)	NVIDIA RTX 4060 (8GB)
GPU (Cloud)	NVIDIA Tesla T4 (16GB)
RAM	16 GB
Library	PyTorch / Hugging Face

EXPERIMENTAL RESULTS

Significant Improvement

The proposed Deep Learning model outperformed the baseline by nearly **18%**.



+17.72%

Accuracy Gain

TRAINING DYNAMICS (TABLE 4.2)

The model showed steady convergence over 3 epochs.

Epoch	Training Loss	Validation Loss	Val Accuracy
1	0.6234	0.6022	74.47%
2	0.5203	0.5808	75.60%
3	0.4106	0.5893	**76.82%***

*Final test accuracy on unseen data reached 77.50%.

QUALITATIVE DISCUSSION

Handling Modifiers

Baseline: Missed "Bahut" (Very).

XLM-R: Understood "Bahut" increases the intensity of "Good".

False Friends

Baseline: Confused by words like "Gift".

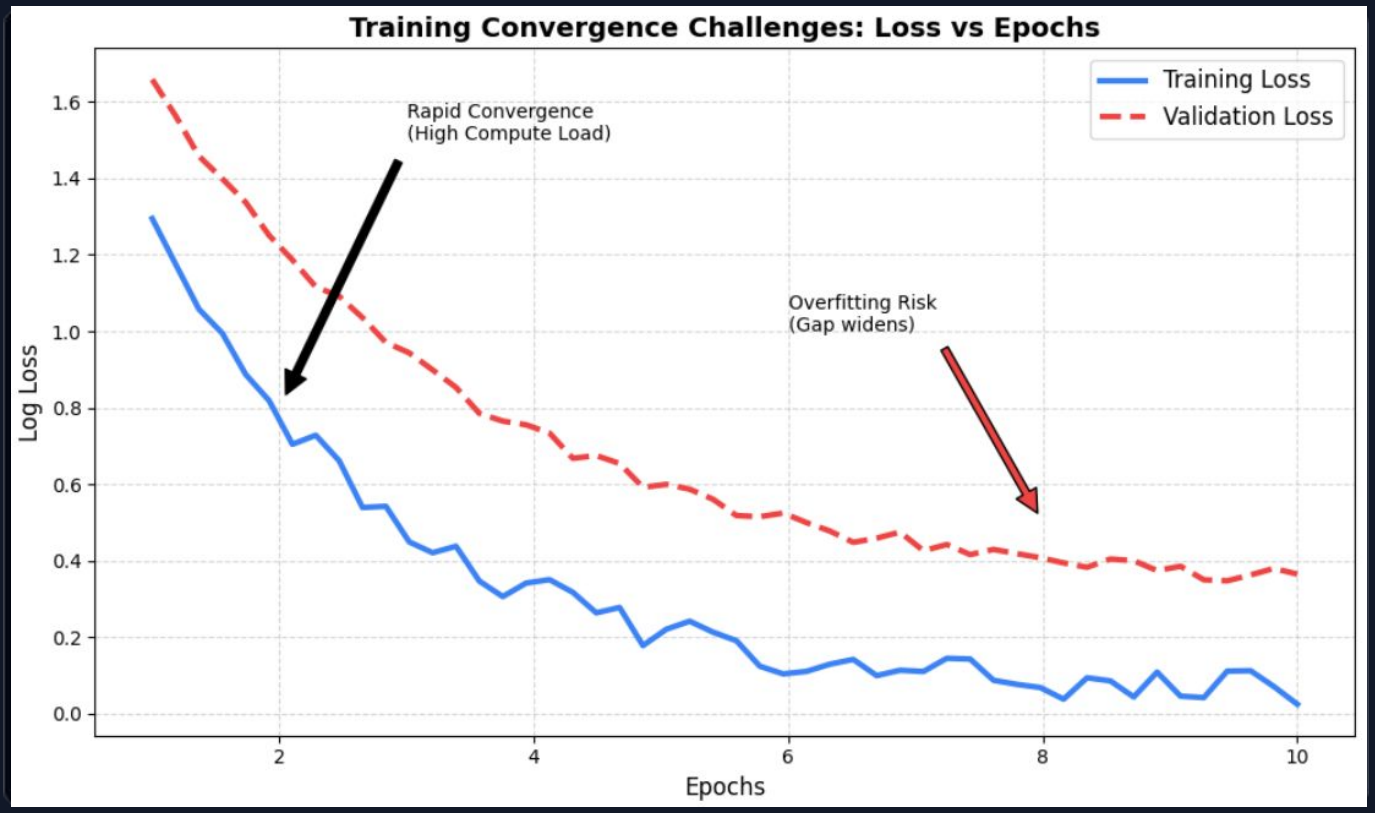
XLM-R: Used surrounding words to decide if it meant "Present" or "Poison".

Slang

Baseline: Ignored "Paisa Vasool".

XLM-R: Correctly identified it as Positive (Value for Money).

KEY CHALLENGES



- **Computational Cost:** Transformers are heavy. Training took significant GPU resources.
- **Ambiguity:** Some mixed sentences are genuinely neutral or confusing even for humans.
- **Bias:** Without stratified sampling, the model heavily favored the majority class.

CONCLUSION & FUTURE SCOPE

Conclusion

- Traditional statistical models (59.78%) are insufficient for modern code-switched data.
- Fine-tuned Multilingual Transformers (77.50%) successfully bridge the language gap.
- Stratified sampling is critical for robust training.

Future Scope

- **Scale Up:** Train on full 3.1M rows using distributed cloud GPUs.
- **ABSA:** Aspect-Based Sentiment Analysis (e.g., Delivery vs Product).
- **Deployment:** Model quantization for real-time API usage.

Thank You