Nepali Sentiment Analysis of Post-COVID Data

Using XLMRoberta for Text Classification

Amulya Bhandari   Sailesh Dahal   Sarayu Gautam   Tohfa Niraula

May 2, 2025

Department of Computer Engineering
Kathmandu University

Outline

Introduction

What is Sentiment Analysis?

- Sentiment analysis classifies text based on emotion or opinion.
- Categories:
    - Positive — praise, approval
    - Neutral — factual
    - Negative — criticism, disapproval
- Applications:
    - Social media monitoring
    - Product reviews
    - Survey analysis

Problem Statement

What Are We Solving?

- Goal: Classify Nepali-language text into sentiment categories.
- Motivation:
  - Nepali is underrepresented in NLP.
  - Enhance Accessibility for Nepali Language.
- Objectives:
  1. Clean and preprocess post-COVID Nepali data.
  2. Train a multilingual BERT model.
  3. Evaluate performance using real-world test data.

Dataset Description

About the Dataset

- Source: Nepali COVID/post-COVID text samples.
- Total Samples:
  - Training: 33,602 samples
  - Testing: 8,401 samples
- Labels: 0 = Negative, 1 = Positive, 2 = Neutral
- Common issues:
  - Invalid labels ('o', '-', etc.)
  - Missing values and noisy characters

	text	label
0	कोभिड बारे हालसम्मको विकास क्रम	0
1	नेताहरु भ्रष्टाचार गर्छन जनताको छोराछोरी बिदेश...	0
2	गौतमबुद्द अन्तराष्ट्रिए क्रिकेट स्टडिएमको नराम...	1
3	दाइ हजुरको भिउज किन कम आज भोलि	0
4	कोभिड नेपालमा जिडिपीको प्रतिशतसम्म क्षति हुनसक्ने	0

Figure 1: Data Sample

Data Preprocessing

Data Cleaning Steps

Steps we took:

1. Removed missing and malformed data.
2. Filtered invalid labels.
3. Tokenized using XLM-Roberta tokenizer.
4. Truncated inputs to max length of 256 tokens.

Result: Clean, structured datasets ready for training/testing.

	text
count	33602.000000
mean	30.863490
std	21.971628
min	1.000000
25%	15.000000
50%	25.000000
75%	43.000000
max	1428.000000

Figure 2: Data Length

Tokenization and Encoding

Tokenizing with XLM-Roberta

- XLM-Roberta tokenizer is multilingual and supports Nepali language.
- Key features of `tokenizer.batch_encode_plus`:
  - Automatically handles padding and truncation for consistent input length.
  - Generates:
    - `input_ids`: Numerical representation of tokens.
    - `attention_mask`: Identifies non-padded tokens for focus.
- Attention masks ensure the model processes only relevant tokens. It tells the model which sentence is real and which is padding.

Understanding Tokenization and Encoding

- Tokenizing is like chopping a sentence into Lego blocks that the model knows.
- Encoding is turning those blocks into numbers so the model can do math with them.
- Why It Matters:
    - The model sees only numbers.
    - Tokenization keeps input size fixed (e.g., max_length=256). Long texts are trimmed, short ones are padded.

Tokenization Example

Step	Output
Input Sentence	सरकारले अस्पतालमा निःशुल्क उपचार उपलब्ध गरायो।
Tokens	['▁सरकारले', '▁अस्पतालमा', '▁नि', 'ः', 'शुल्क', '▁उपचार', '▁उपलब्ध', '▁गरायो', '।']
Token IDs	[25033, 149007, 946, 6, 153794, 26409, 38071, 87995, 4]

- The model doesn't understand text. It only understands numbers.
- Tokenizer breaks the sentence into subwords it knows (like ‘नि’, ‘ः’, ‘शुल्क’ for ‘निःशुल्क’).
- The tokenizer assigns a unique ID to each known token.
- These token IDs are what get passed to the model as input.

Figure 3: Tokenization Example

Model Architecture

XLM-Roberta Model Details

Transformers process all words at the same time (not one-by-one like RNN).

Structure:

- XLM-R supports multiple languages, including Nepali, out of the box.
- Fine-tuning was performed to adapt the model's top layers specifically for sentiment classification.
- Produces probabilities for three sentiment classes: Negative, Positive, and Neutral.

Trained on: PyTorch with mixed precision (autocast enabled)

Model Usage Example

"यो उत्पादन उत्कृष्ट छ।"

Tokenized Input:

- Tokens: [CLS] यो उत्पादन उत्कृष्ट छ । [SEP]
- Converted to input_ids: [101, 2345, 5678, 9101, 1123, 102] (Note: These are example token IDs.)

Model Output (Softmax Probabilities):

- Negative: 0.05
- Neutral : 0.10
- Positive: 0.85

Prediction: → Positive sentiment

- [CLS] and [SEP] are special tokens used by BERT-like models (e.g., XLM-Roberta).
- input_ids are the numerical representation of tokens.
- The model outputs a probability score for each class.
- The highest score (0.85 for Positive) becomes the final prediction.

Figure 4: Model Usage Example

Training Pipeline

Main Workflow Steps

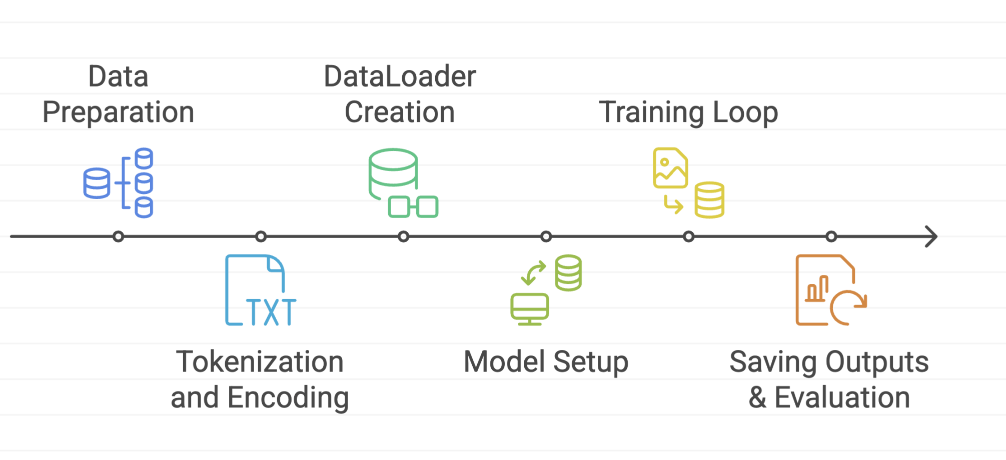

Figure 5: Workflow Steps

Training Loop

Figure 6: Training Loop

Deployment and Evaluation

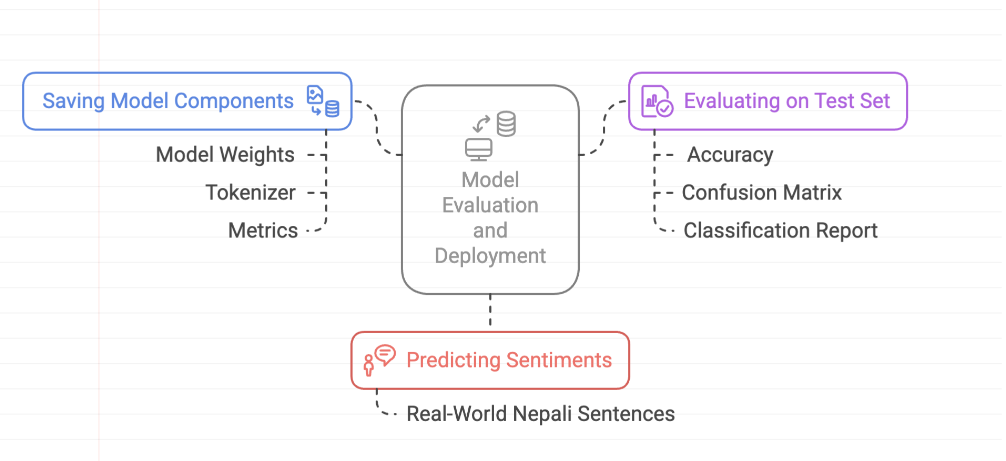

Figure 7: Deployment

Results

Loss and Accuracy Over Epochs

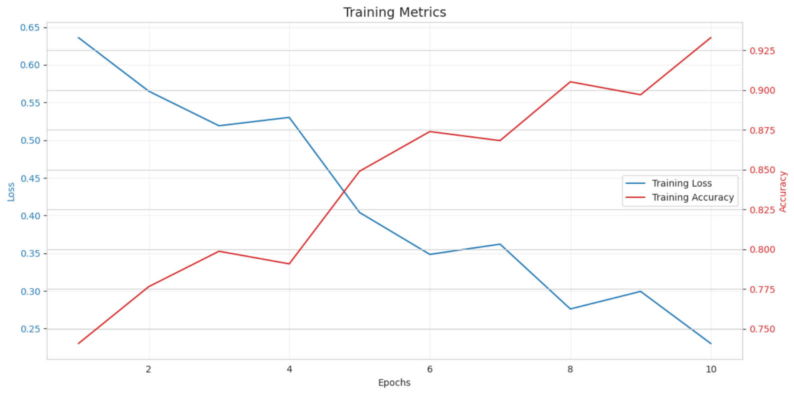

Figure 8: Loss and Accuracy

Test Set Evaluation Metrics

Label	Precision	Recall	F1-score
Negative (0)	0.80	0.74	0.77
Positive (1)	0.78	0.83	0.80
Neutral (2)	0.52	0.52	0.52
Overall Accuracy			**74.0%**

Key Insights:

- High precision/recall for Positive/Negative.
- Neutral class more ambiguous $\rightarrow$ lower performance.

Model Performance on Test Set

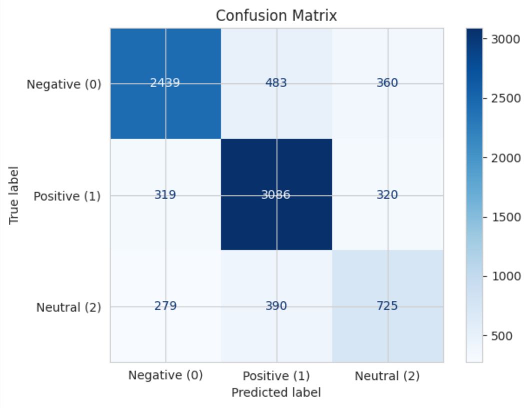

Figure 9: Confusion Matrix

	prec	recall	f1	support
Neg (0)	0.80	0.74	0.77	3282
Pos (1)	0.78	0.83	0.80	3725
Neut (2)	0.52	0.52	0.52	1394
accuracy			0.74	8401
macro avg	0.70	0.70	0.70	8401
weighted avg	0.74	0.74	0.74	8401

Table 1: Classification Report

Sample Predictions on Unseen Data

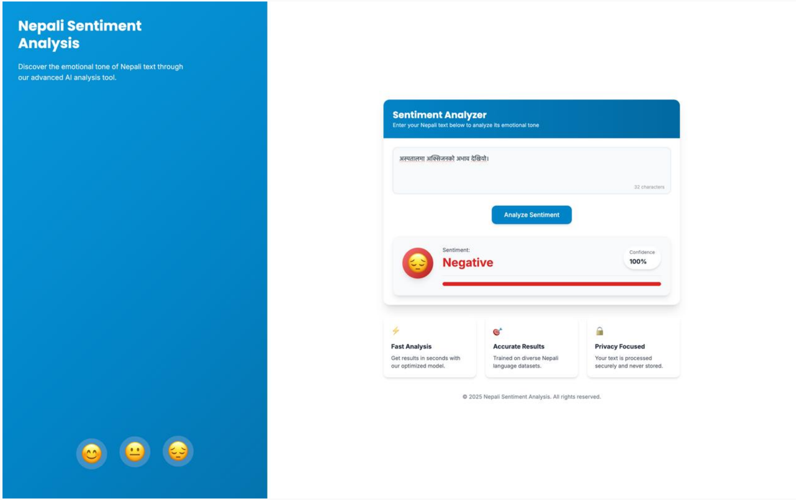

Figure 10: Sample Prediction

Conclusion

Conclusion and Future Work

Key Takeaways:

- Trained a sentiment classifier on Nepali-language text using XLM-Roberta.
- Achieved 74% of overall accuracy.
- Strong performance on binary sentiment; neutral remains challenging.

Future Improvements:

1. Larger or augmented datasets.
2. Additional validation set for tuning.
3. Model deployment as an API/web service.

Thank You!

Questions or feedback?

Project Resources:

GitHub: `github.com/saileshbro/ai-proj`

We appreciate your time and attention!