Nepali Sentiment Analysis of Post Covid Data

Using BERT for Text Classification

Amulya Bhandari   Sailesh Dahal   Sarayu Gautam   Tohfa Niraula

April 24, 2025

Department of Computer Engineering
Kathmandu University

Outline

Introduction

Problem Statement

Dataset Description

Data Preprocessing

Tokenization and Encoding

Model Architecture

Training Pipeline

Challenges Faced

Future Work

Introduction

What is Sentiment Analysis?

- Sentiment analysis is an NLP task that classifies text based on emotion or opinion.
- Common categories:
  - Positive — praise, agreement
  - Neutral — factual or no emotion
  - Negative — criticism, disagreement
- Applications:
  - Product reviews
  - Social media monitoring
  - Survey analysis

Problem Statement

What Are We Solving?

- Goal: Classify Nepali-language text into sentiment categories.
- Why Nepali?
  - Underrepresented in NLP research
  - Fewer labeled datasets and tools available
- Focus of project:
  1. Clean and preprocess Nepali text data
  2. Train a multilingual BERT model
  3. Evaluate classification performance

Dataset Description

About the Dataset

- Source: Two CSV files — train.csv and test.csv
- Each row includes:
  - Text — sentence in Nepali
  - Label — 0 (Negative), 1 (Positive), 2 (Neutral)
- Data issues:
  - Missing values and malformed entries
  - Invalid labels such as ”-”, ”o”
  - Non-standard characters or encoding problems

Data Preprocessing

Cleaning and Preparing the Data

Steps we took to clean the dataset:

1. Dropped rows with missing or empty text.
2. Removed invalid labels.
3. Converted label strings to integers.
4. Tokenized text using a pretrained BERT tokenizer.

Result: A clean, structured dataset suitable for training.

Tokenization and Encoding

Using BERT Tokenizer

Why tokenization?

- Machine learning models require numerical input.
- Tokenizer converts words/subwords into integer IDs.

In our project:

- Tool used: Hugging Face's tokenizer (bert-base-multilingual-cased)
- Features:
  - Handles over 100 languages including Nepali
  - Supports padding/truncation (max length: 512)
  - Generates attention masks for input sequences

Model Architecture

BERT for Sequence Classification

Model details:

- Base: Pretrained BERT model from Hugging Face Transformers
- Head: Fully connected classification layer with softmax activation
- Output: Probability distribution over three classes

Why use BERT?

- Captures contextual meaning using attention mechanisms
- Multilingual support makes it suitable for Nepali text

Training Pipeline

Training Configuration

Important training settings:

- Optimizer: AdamW (with weight decay)
- Learning rate: 2×10^{-5}
- Batch size: 16
- Epochs: 10
- Loss function: Cross-entropy loss

Training implementation:

- PyTorch framework with GPU support (if available)
- DataLoader used for efficient batching and shuffling

Challenges Faced

What Were the Difficulties?

Challenges encountered during development:

1. Limited dataset size → overfitting risk
2. Noisy labels → needed extensive cleaning
3. Imbalanced classes → biased model predictions
4. No validation set → difficult to monitor performance during training

Future Work

Next Steps and Improvements

Planned improvements:

1. Add a proper validation set for tuning.
2. Use metrics like F1-score and confusion matrix.
3. Try models like XLM-Roberta or mMiniLM.
4. Apply hyperparameter tuning and early stopping.
5. Deploy the model via API or web interface.

Thank You!

Questions or feedback?

Project Resources:

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

We appreciate your time and attention!