HINDI MULTI-CLASS EMOTION DETECTION

REVIEW - 3

Team:
G. Sai Shivananda(22BCE7263)
K. MADHUKUMAR (22BCE9883)
M. BHANU SIVA RAM
(22BCE9724)

Abstract

This project addresses automatic emotion detection in Hindi mental-health text, leveraging deep learning and advanced language representation. We utilize the BHAAV dataset, a large Hindi emotion corpus annotated for five classes: anger, joy, sadness, surprise, and neutral. Preprocessing involves custom Hindi stopword removal, tokenization, and class balancing using SMOTE. Each sentence is converted into a high-dimensional vector using mBERT embeddings. These embeddings serve as input to a BiLSTM neural classifier designed to capture both forward and backward context in text. The model achieves robust performance with over 90% accuracy and macro-F1 score, despite challenges of class imbalance and nuanced emotional overlap. Results are deployed as a Streamlit web app, allowing real-time emotion prediction for Hindi sentences. This work demonstrates an effective, scalable solution for emotion analysis in low-resource languages and provides a practical tool for psychological and sentiment applications in Hindi NLP.

Introduction

- Humans are the only creatures with true intelligence.
- What makes humans unique is their emotional intelligence, the ability to understand, feel, and respond to emotions.
- Emotional intelligence enables empathy, connection, and deeper decision-making.
- Imagine how impactful it would be to build machines with emotional intelligence.
- Machines with emotional intelligence could understand and respond to human emotions sensitively.
- This would transform human-computer interaction into more meaningful, supportive, and intuitive experiences.

Need

- Many people express their feelings in Hindi online, but most technology only works well with English.
- Hindi voices and emotions are often ignored by existing tools.
- There is a gap for emotion analysis technology made specifically for Hindi, which can be helpful in counseling, mental health, and daily conversations.
- Making this technology widely available helps more people feel heard and understood.

Objectives

- Build a reliable tool that can recognize and classify key emotions—anger, joy, sadness, surprise, and neutral—from Hindi sentences.
- Make emotion analysis technology accessible and accurate for Hindi speakers, addressing the gap left by English-centered tools.
- Use advanced models and data techniques to improve how computers understand the emotional meaning in Hindi text.

Existing Systems

BHAAV – A Text Corpus for Emotion Analysis from Hindi Stories

- Models tested: SVM, Logistic Regression, Random Forest, CNN, BiLSTM.
- Best result: Logistic Regression (macro-F1 0.58, accuracy 0.62).
- Main issue: Neutral sentences much more common–strong class imbalance.

EmoInHindi: Multi-label Emotion and Intensity in Dialogues

- Hindi dialogues labeled for multiple emotions (16 types, intensity).
- Used BiLSTM, CMN, and contextual Transformer models.
- Best (Transformer): 67% accuracy, 66% F1-score.
- Data mostly from counseling/legal domains, annotation is complex.

Existing Systems

Multi-label Emotion Classification of Urdu Tweets

- Dataset: 6,043 Urdu tweets, multi-label for six emotions.
- Models: Random Forest (best, accuracy 51.2%, macro-F1 56%), SVM, CNN, LSTM, BERT.
- Deep models performed worse than Random Forest—BERT had no advantage due to limited Urdu resources.

Emotion Detection Framework for Twitter Data Using Supervised Classifiers

- Used Naive Bayes and KNN for emotion detection on tweets (four emotion categories).
- Naive Bayes outperformed KNN: accuracy 73%.
- Approach is rule-based labeling plus classic ML—no deep learning

Methodology

Data Preprocessing

- Used BHAAV Hindi dataset, 5 emotion classes.
- Cleaned, tokenized sentences, removed stopwords.

Feature Extraction

- Generated mBERT embeddings for each sentence.

Data Balancing

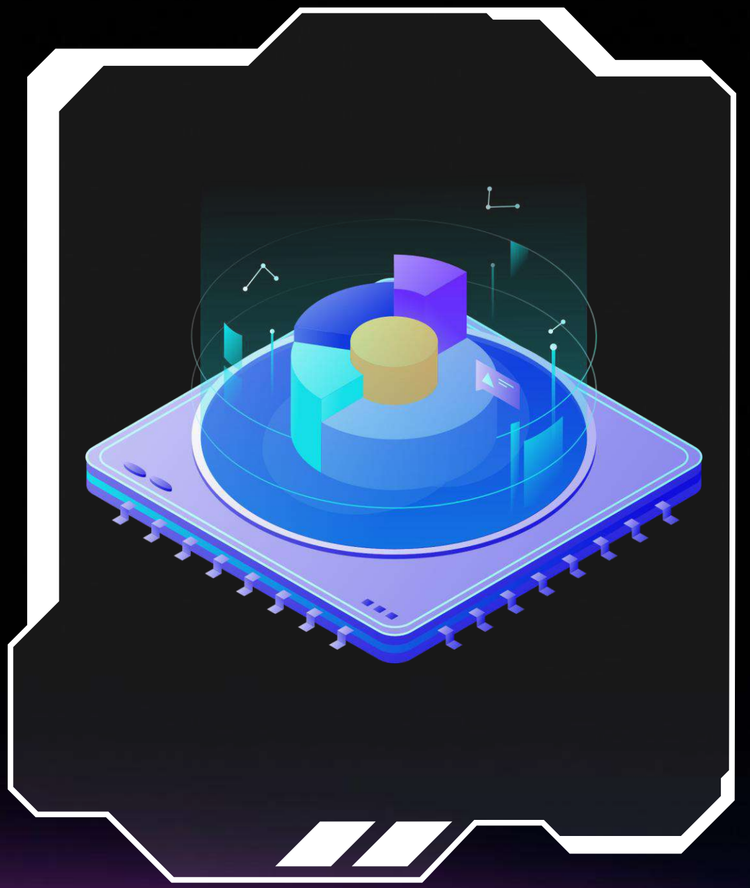
- Applied SMOTE to handle class imbalance

Model Training

- Built BiLSTM neural network with dropout and dense layers

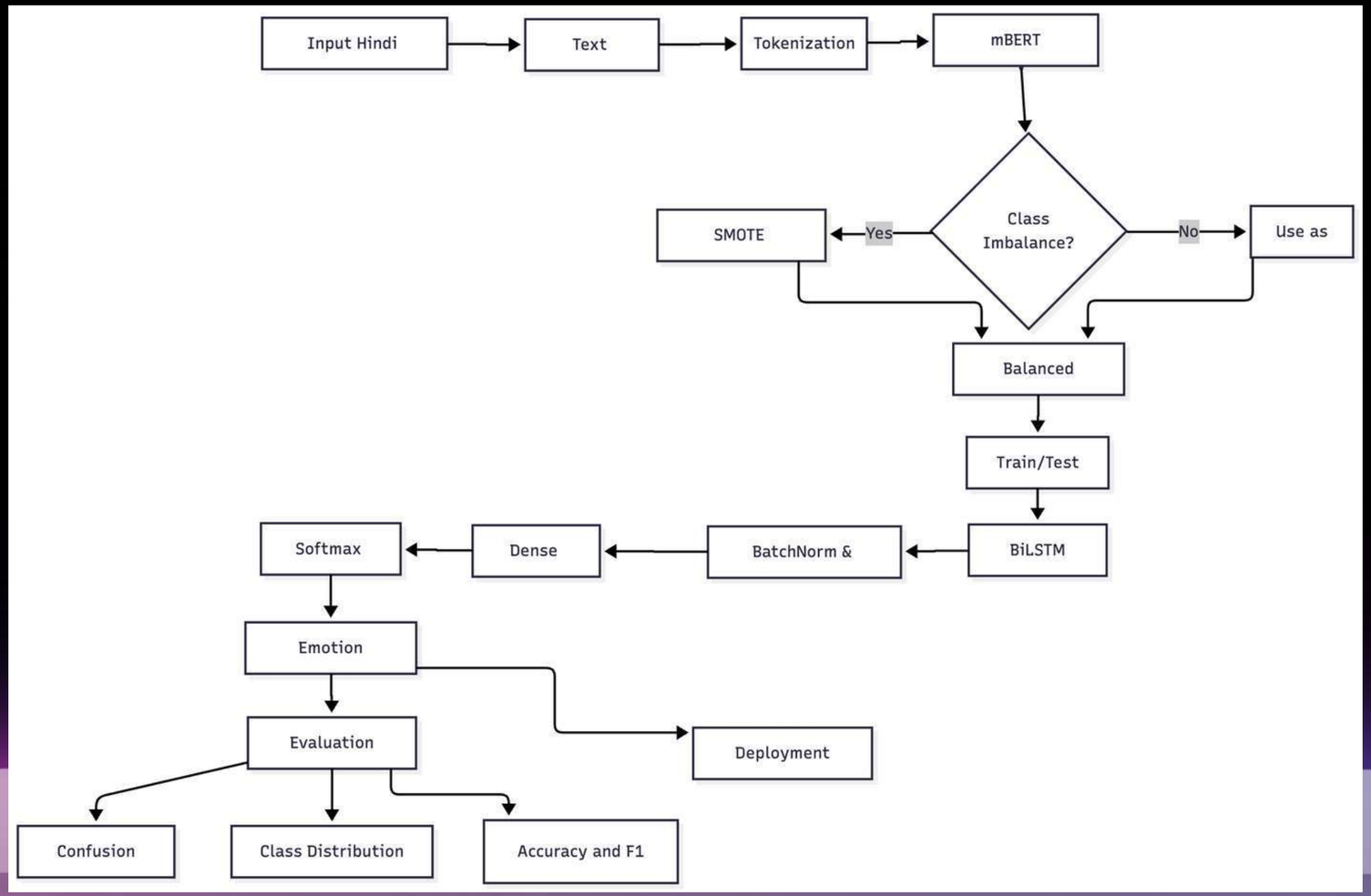
Evaluation

- Measure Accuracy, Precision, Recall, F1-score.



Methodology

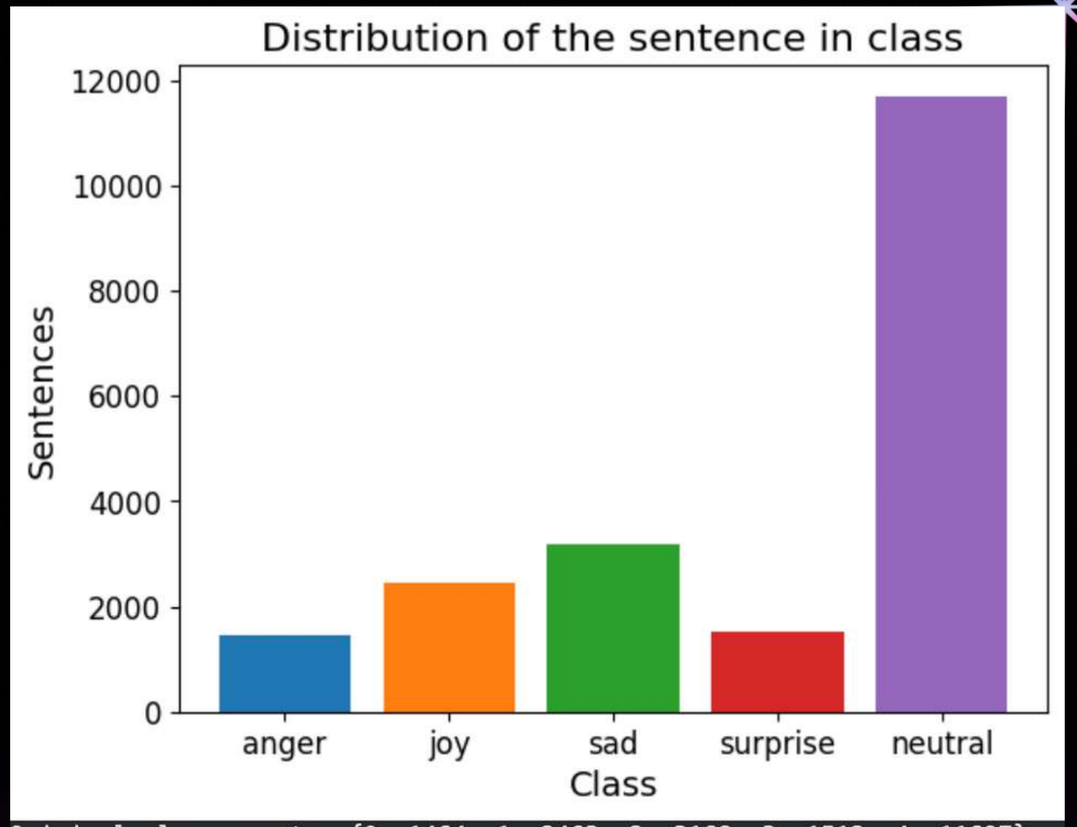
Architecture



Methodology

Dataset

- BHAAV dataset: 20,000+ Hindi sentences, each labeled with one of five emotions (anger, joy, sadness, surprise, neutral).
- Publicly available, annotated by native speakers for high quality.
- Used to train and test emotion classification models—covers diverse genres and real-world language.




```
Original class counts: {0: 1464, 1: 2463, 2: 3168, 3: 1512, 4: 11697}
```

Methodology

Text Processing

1. Cleaning
   - All sentences are converted to string and unwanted characters (like punctuation, extra spaces, special symbols) are removed using regular expressions.
   - Example:
     - Before cleaning: "यह फिल्म, वास्तव में- अति&#$@ सुंदर है!"
     - After cleaning: "यह फिल्म वास्तव में अति सुंदर है"

2. Tokenization
   - Used the indicnlp library (from indicnlp.tokenize import indic_tokenize) to tokenize Hindi text at word level.
   - Example:
     - Original: "यह फिल्म वास्तव में सुंदर है"
     - Tokenized: ["यह", "फिल्म", "वास्तव", "में", "सुंदर", "है"]

Methodology

Text Processing

3. Stopword Removal
   - Downloaded a Hindi stopword list (from stopwords-iso GitHub).
   - Filtered tokens, removing any word present in the stopword list (hindistopwords set).
   - Example:
     - Before stopword removal: ["यह", "फिल्म", "वास्तव", "में", "सुंदर", "है"]
     - After stopword removal: ["फिल्म", "सुंदर"]

	Sentence
13855	इतना सुनते ही तेजी से भागी अपने कमरे की तरफ और...
19470	इसलिए जहाँ उसकी मण्डली के और लोग गाँव के सरगना...
1009	झुकी हुई कमर, पोपला मुँह, सन के–से बाल–इतनी सा...
18238	स्वाधीनता आंदोलन के दौरान वे सात बार जेल गए
4641	भला पंचायत की बात कैसे टाल सकता था साहूकार।

	Sentence_Clean
13855	इतन सुनत तेज भाग कमर तरफ तकिय उठ देख 100 रुपए ...
19470	इसलिए उसक मण्डल लोग गाँव सरगन मुखिय बन , गाँव ...
1009	झुक कमर , पोपल मुँह , सन – बाल – इतन सामग्र एक...
18238	स्वाधीनत आंदोलन दौरान सात बार जेल गए
4641	भल पंचायत बात कैस टाल साहूकार ।

Methodology

Feature Extraction

- Used transformers library, specifically bert-base-multilingual-cased from Hugging Face.
- Each cleaned Hindi sentence is tokenized, padding/truncating to a fixed length (50 tokens).
- Sentences are passed through mBERT to generate embeddings:
  - The last hidden state is averaged to get a 768-dimensional vector per sentence.
- Embedding process:
  - Example sentence: "फिल्म सुंदर"
  - Tokenized + encoded → mBERT embedding: [0.13, -0.22, ..., 0.07] (length: 768)
- These embeddings are used as input features for model training.

Why mBERT (multilingual BERT), not BERT?

- Standard BERT only supports English.
- mBERT is trained on 104 languages (including Hindi), so it works for Hindi text.

Methodology

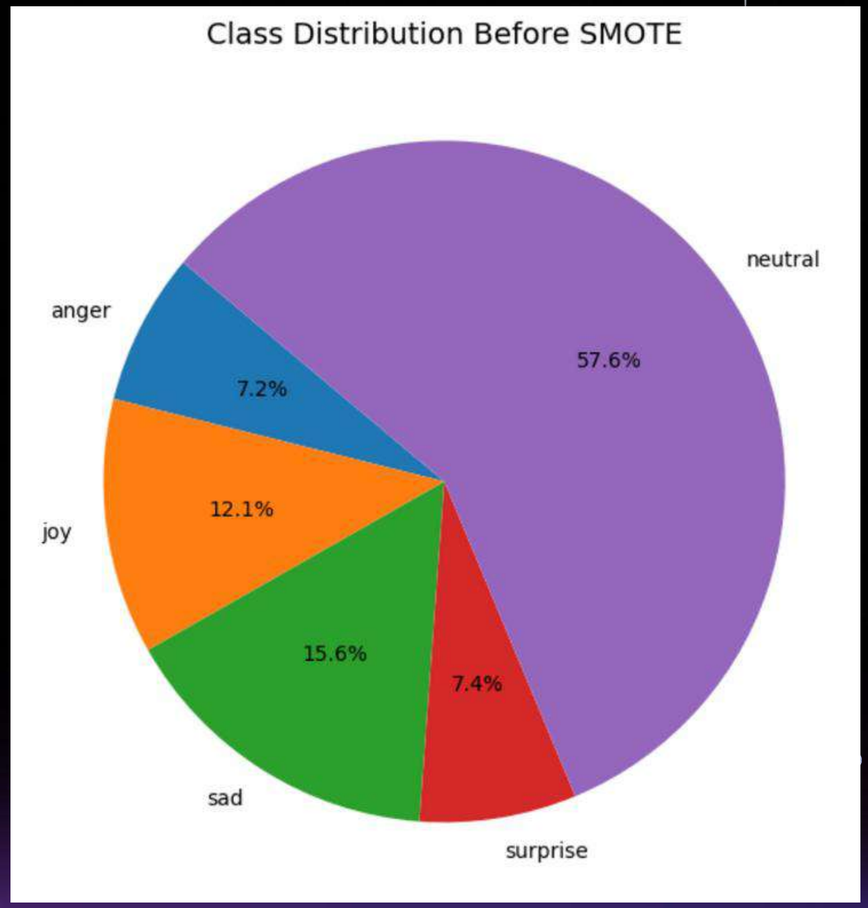
Feature Extraction

- Architecture: Same as original BERT base, 12 Transformer encoder layers, 12 attention heads, hidden layer size 768.
- Uses WordPiece tokenization with a shared 120,000-word vocabulary for all supported languages.
- Trained with Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) objectives on mixed-lingual input.
- Each input sequence (words or subwords) → Token+Position+Segment embeddings → Multi-layer bi-directional transformer → [CLS] pooled output or per-token outputs.

Methodology

Data Balancing

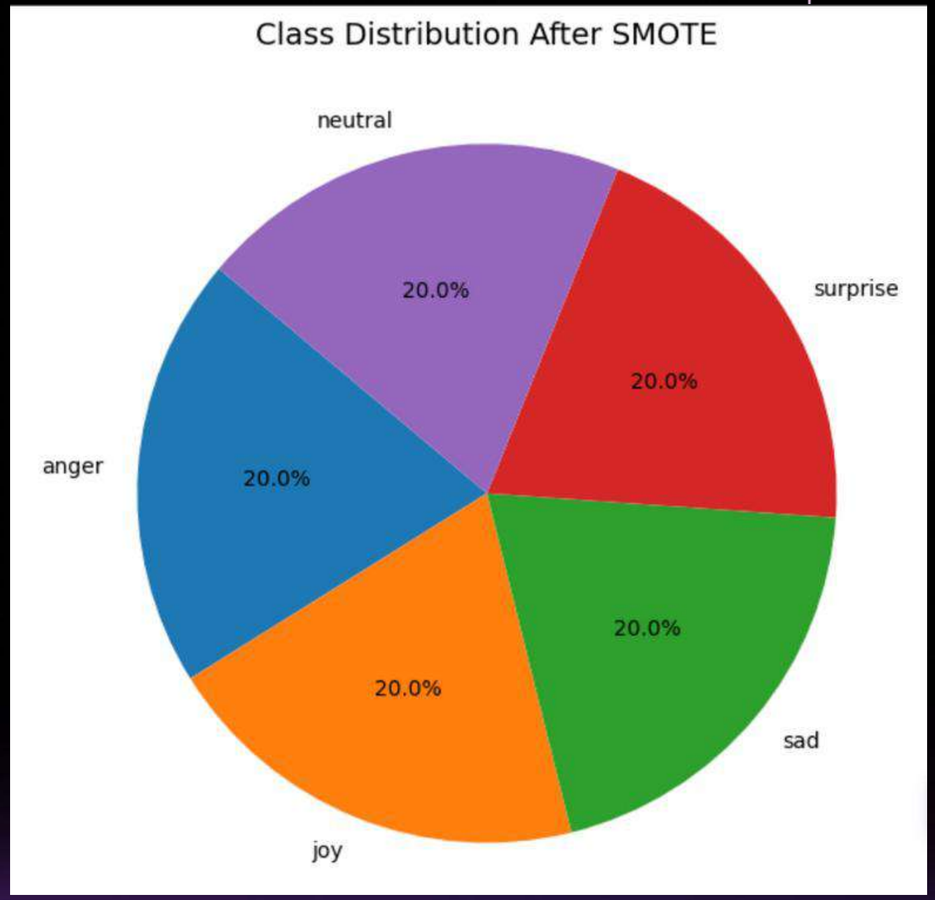
- The dataset is highly imbalanced: classes like "neutral" dominate, while classes like "anger" and "surprise" are rare.
- Imbalanced data makes the model biased toward frequent classes, reducing accuracy for minority emotions.
- Used SMOTE (Synthetic Minority Over-sampling Technique) to oversample minority emotion classes in the BERT embedding space.



Methodology

Data Balancing

- SMOTE makes new data points by looking at two real examples and making a new one "in between" them (just as you could find a point between two dots on a line).
- Before SMOTE: Pie/barcharts show "neutral" class with 11,697 samples; "anger" only 1,464; "surprise" 1,512, etc.
- After SMOTE: All classes balanced to 11,697 samples; class distribution plots become uniform.



Methodology

BiLSTM

- A Bidirectional LSTM (BiLSTM) is a recurrent neural network used primarily on natural language processing.

- It is an extension of the traditional LSTM (Long Short-Term Memory) network.

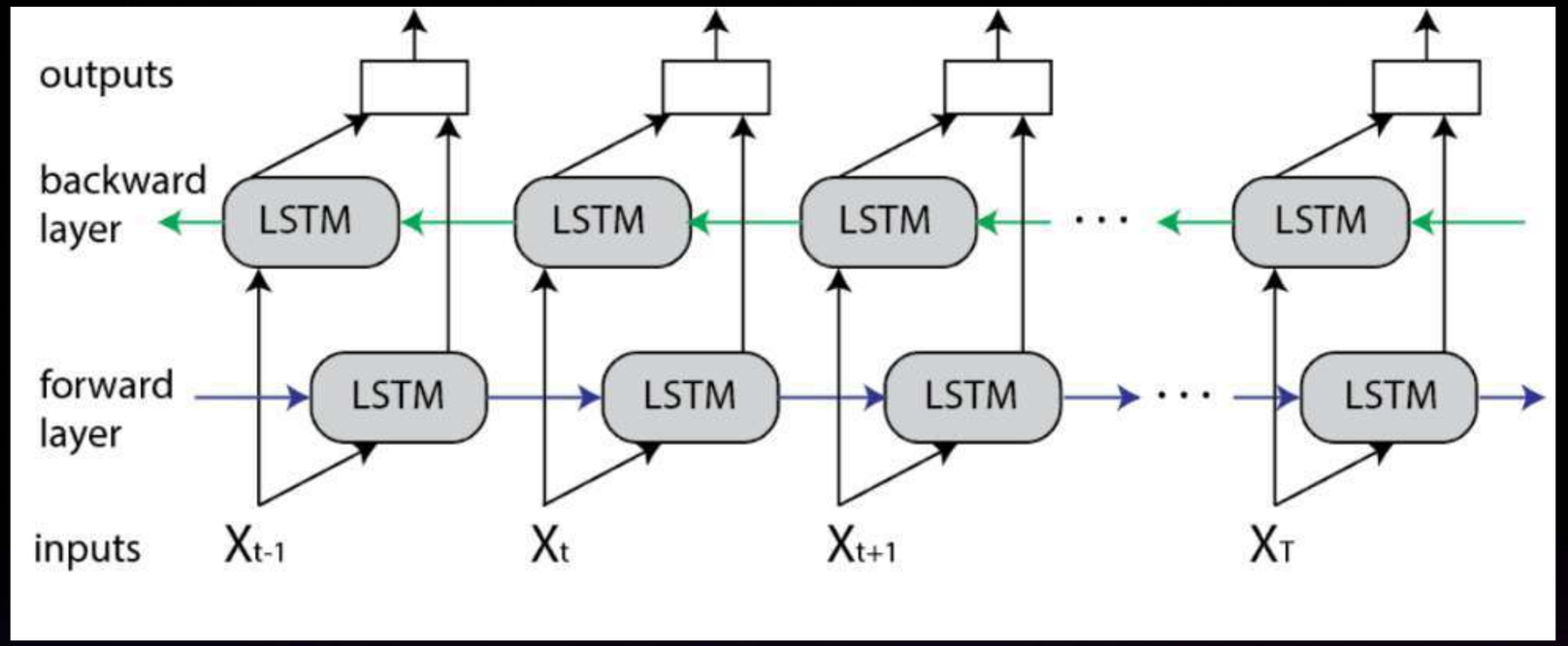
- BiLSTMs allow information to flow from both forward and backward enabling them to capture richer contextual information.

- A Bidirectional LSTM (BiLSTM) consists of two separate LSTM layers:

1. Forward LSTM: Processes the sequence from start to end
2. Backward LSTM: Processes the sequence from end to start

Methodology

BiLSTM Architecture



Methodology

How BiLSTM works?

1. Input Layer

- Takes in the sentence embedding (from mBERT)
- Each value in this vector is a numerical feature representing word/sequence information in the sentence.

2. Forward Pass

- Processes the input data from first to last feature index, updating hidden state at each time step.

3. Backward Pass

- Processes the input data from last to first feature index, capturing dependencies from future context.

Methodology

How BiLSTM works?

4. Combining Outputs

- After the whole sequence is processed forwards (and backwards), you get two outputs per word/step: one from left-to-right history, and one from right-to-left history.
- These outputs are concatenated (put side by side) to make a richer vector for each step.

5. Dropout Layer

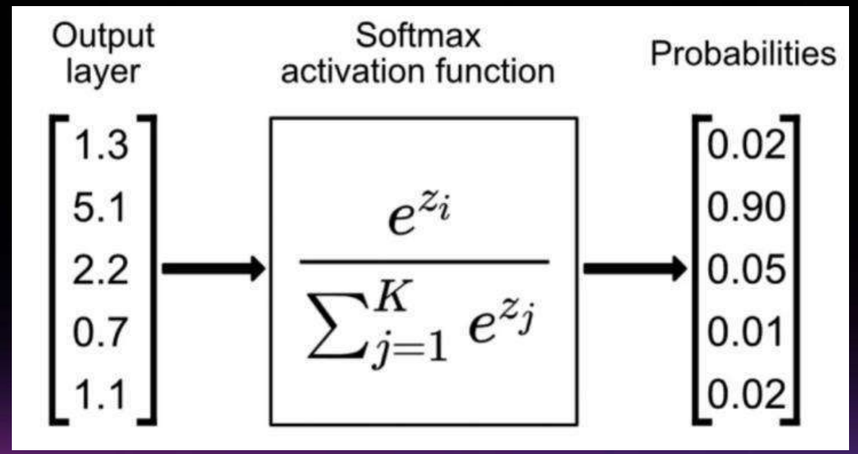
- Some values in the output vector are randomly “turned off” during training (e.g., with rate=0.5, half are set to zero).
- This makes the model learn features that generalize well and aren't just memorized patterns.

Methodology

How BiLSTM works?

6. Softmax layer

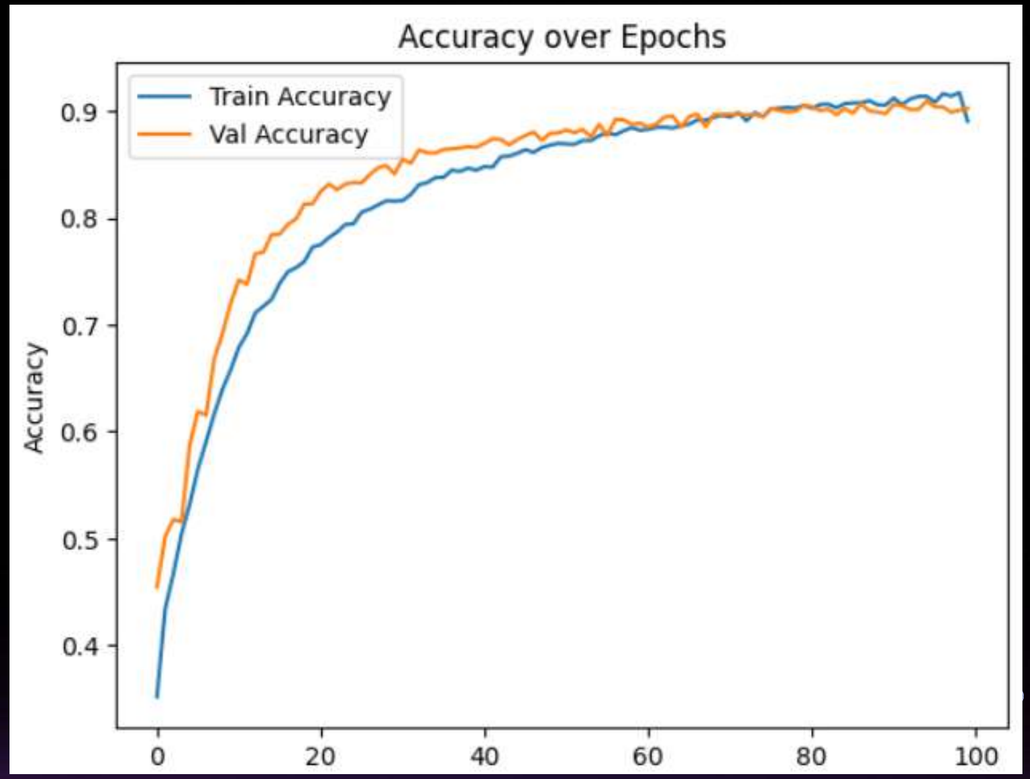
- The Dense layer compresses your BiLSTM features into one number (logit) per class, using weights it learned.
- Softmax takes the raw output scores (logits) from the last layer of your model and turns them into probabilities for each class.



Results

Accuracy over Epochs

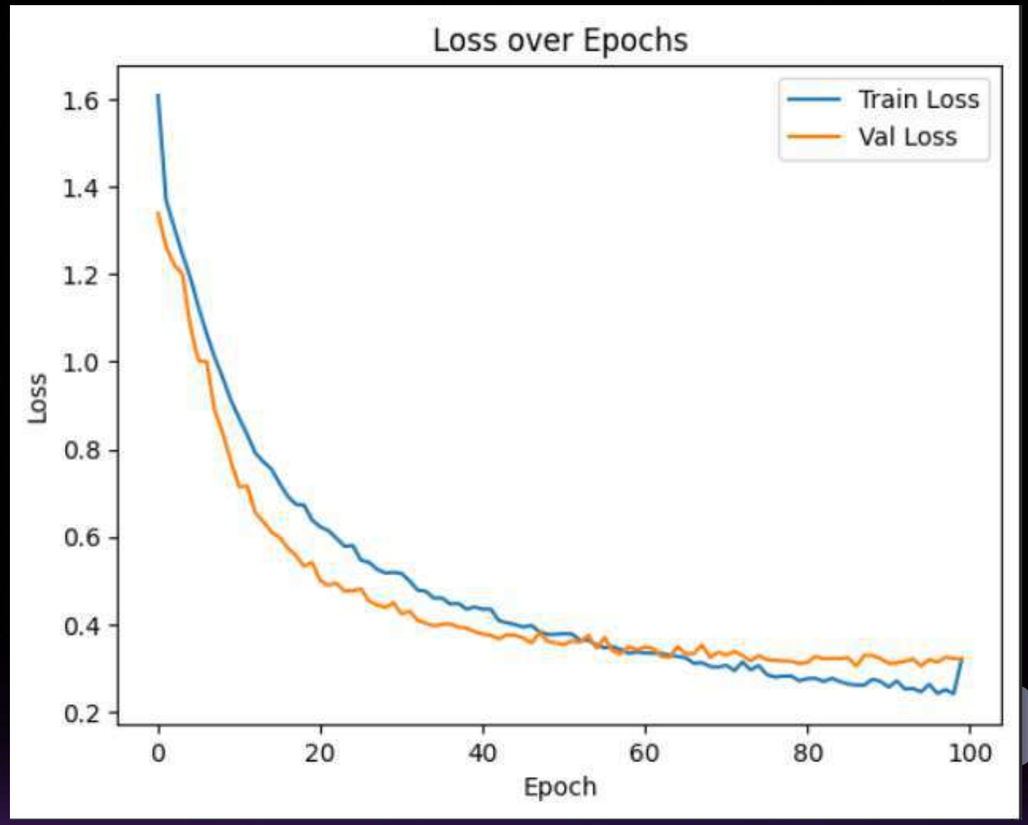
- Shows how well the model learns to predict correct emotion labels during training.
- Training accuracy gradually increases, indicating successful learning.
- The upward trend indicates the model is continuously improving and not underfitting.
- Minimal separation between train & validation accuracy suggests limited overfitting.



Results

Loss over Epochs

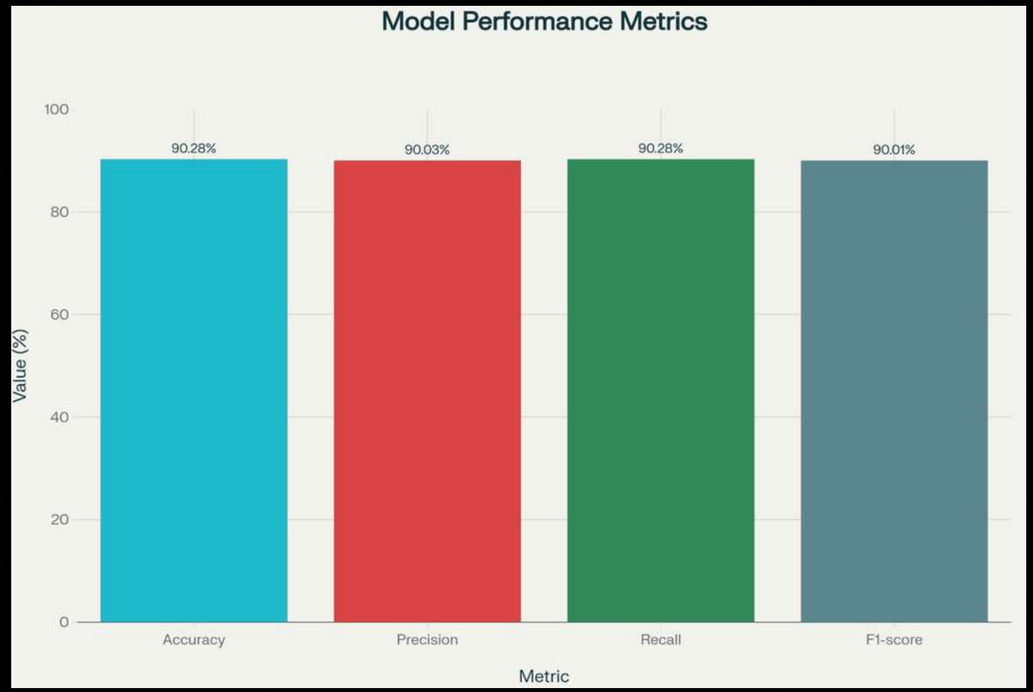
- Training loss decreases steadily, showing reduced prediction errors.
- Validation loss also declines, confirming stable and effective learning.
- Loss curve becomes flatter later, showing convergence.
- Declining loss indicates improved model confidence, suggesting good optimization and controlled overfitting.



Results

Overall Metrics

- Represents percentage of correctly predicted labels on the test dataset.
- Achieved high accuracy, indicating strong performance with multilingual embedding + BiLSTM.
- The accuracy metric highlights that the model is reliable for real-world sentiment classification in Hindi text.




```
Overall Accuracy: 0.9028
Macro Precision:  0.9003
Macro Recall:     0.9028
Macro F1-score:   0.9001
```

Results

Classification Report

- Includes precision, recall, and F1-score for each emotion class.
- Shows balanced performance across classes after SMOTE balancing.
- High F1-scores mean the model performs consistently for all classes.
- Lower score in some classes may indicate confusion due to subtle text differences.

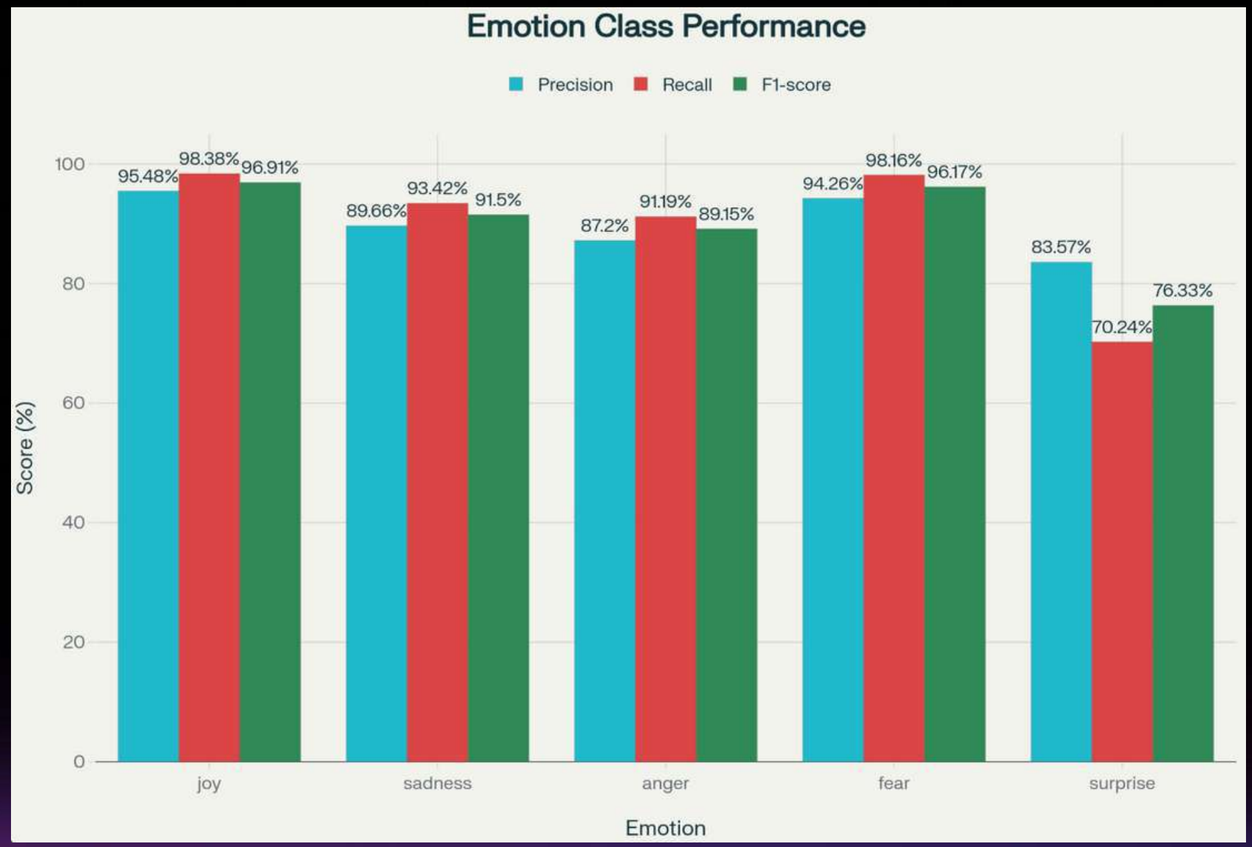
Classification Report:

	precision	recall	f1-score	support
0	0.9548	0.9838	0.9691	2340
1	0.8966	0.9342	0.9150	2339
2	0.8720	0.9119	0.8915	2339
3	0.9426	0.9816	0.9617	2340
4	0.8357	0.7024	0.7633	2339
accuracy			0.9028	11697
macro avg	0.9003	0.9028	0.9001	11697
weighted avg	0.9003	0.9028	0.9001	11697

Results

Class-wise Metrics

- Evaluates precision, recall, and F1 for each label (Anger, Joy, Sadness, Surprise, Neutral).
- Indicates which emotions the model classifies best.
- Helps identify improvement areas (e.g., similar emotions might be confused).
- “Joy & Neutral” may have higher scores due to clearer textual cues, while “Surprise or Anger” might show lower recall due to overlapping expressions.



Hardware & Software requirements

Hardware

- Processor: Intel i5 / AMD equivalent or higher
- RAM: Minimum 8 GB
- Storage: 256 GB HDD / SSD
- GPU: NVIDIA CUDA-enabled GPU (optional, for faster model training)

Software

- Operating System: Windows / Linux / Mac OS
- Programming Language: Python 3.8+
- IDE/Notebook: Colab / Jupyter NB / VS Code
- Libraries: Scikit-learn, NLTK, TensorFlow, Pandas, NumPy, Matplotlib, Hugging Face Transformers
- Dataset Source: Twitter, reddit, research corpora

Conclusion

- The project successfully uses advanced machine learning techniques to detect emotions in Hindi text.
- By applying mBERT embeddings, data balancing methods, and BiLSTM networks, our system can classify five main emotions accurately.

THANK YOU!