National College of Ireland

A Deep Learning Emotion Classification Framework for Low Resource Languages

MSc Research Project
MS in Data Analytics

Manisha
Student ID: x21207194@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Paul Stynes, William Clifford, Eugene McLaughlin

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Manisha
Student ID:	x21207194@student.ncirl.ie
Programme:	MS in Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Paul Stynes, William Clifford, Eugene McLaughlin
Submission Due Date:	14/08/2023
Project Title:	A Deep Learning Emotion Classification Framework for Low Resource Languages
Word Count:	4445
Page Count:	15

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Manisha
Date:	18th September 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Contents

List of Figures

List of Tables

A Deep Learning Emotion Classification Framework for Low Resource Languages

Manisha
x21207194@student.ncirl.ie

Abstract

Emotion classification from text is the process of identifying and classifying emotions expressed in textual data. Emotions can be feelings such as anger, joy, suspense, sadness, neutral, and so on. Developing a machine learning model to identify emotions in a low-resourced language with a limited set of linguistic resources and annotated corpora is a challenge. This research proposes a Deep Learning Emotion Classification Framework to identify emotions in low-resourced languages such as Hindi. An annotated corpus of Hindi short stories consisting of 20,304 sentences is used to train the models for predicting five categories of emotions: anger, joy, suspense, sadness, and neutral/plain talk. To resolve the class imbalance in the dataset SMOTE technique is applied. The framework leverages fine-tuning pre-trained models, mBERT, IndicBERT, and a hybrid model, mBERT+BiLSTM. In addition, multiple baseline machine learning and deep learning models such as SVM, Logistic Regression, Random Forest, CNN, BiLSTM, and CNN+BiLSTM are experimented with in the research. The results of the models are evaluated based on macro average recall, macro average precision, and macro average F1 score. The hybrid model mBERT+BiLSTM performed best in the experiment with a test accuracy of 57%.

Keywords – deep learning, emotion classification, low resource languages, pre-trained model, transfer learning

1 Introduction

With the increasing accessibility to the internet and the availability of a number of social media platforms, people are spending the majority of their time expressing opinions, communicating feelings, and exchanging ideas on virtual platforms. Comprehending human emotions from these online contents is important for business leaders, social scientists, and other concerned entities. It has a wide variety of applications, such as stress/anxiety retrieval from online chats, detecting sensitive emotions from online conversations, capturing emotions from multimedia tagging, and many more (Acheampong et al. 2020). Detecting emotions from texts is the process of automatically assigning an emotion category to a textual document from a set of predetermined categories (Das et al. 2021). It's a subset of sentiment analysis that aims to find fine-grained emotions such as joy, anger, happy and so on from texts, speech, and even images rather than generic and coarse-grained polarity assignments like neutral, positive, and negative. There is a wealth of text available on social media, customer reviews, and news articles that can be used to gain insights into the emotions of individuals.

A language is termed as low-resourced when it lacks monolingual corpora or linguistic resources required for training AI models. While the development of text classification techniques from high-resource languages like English, Chinese, and French has progressed significantly in recent years, there has been no notable progress in low-resource languages such as Hindi, Bengali, Tamil and Turkey. Today, the majority of the research on natural language processing (NLP) is focused on only 20 of the 7000 global languages leaving the rest understudied (Magueresse et al. 2020). Lack of annotated corpora, extensive monolingual corpora, useful training attributes such as supervised data and limited text processing tools make emotion analysis in low-resource languages a challenge.
Classification based on machine learning (ML) models require manually selecting features from text using techniques like Bag of Words (BOW), and n-grams, whereas deep learning (DL) algorithms can learn features from the text itself, which reduces the need for feature engineering and makes the models reusable across different tasks (Alam et al. 2020). Taking motivation from this, the aim of the research is to investigate to what extent deep learning models can be utilized to classify emotions from low-resource languages like Hindi. The major contribution of this study is a deep-learning emotion classification framework that uses fine-tuning techniques on pre-trained models. Considering the complexities of emotion analysis in resource-constrained languages, this research aims to contribute to the following:

- Investigate the state-of-the-art deep learning models around emotion classification from texts.
- Design a deep learning emotion classification framework and carefully implement it on the dataset.
- Evaluate the model performance based on macro average precision, macro average recall, and macro average F1 score.

This paper discusses the application of deep learning models for emotion classification in low-resourced languages and demonstrates development in the field in Section 2. Sections 3 and 4 explain the methodology, design, and technical aspects of the experiment. Section 5 elaborates on model evaluation results, and section 6 summarizes the outcomes and recommends future work.

2 Related Work

Detecting and analyzing emotions is one of the challenging and emerging areas of natural language processing (NLP). Moreover, the progress for low-resourced languages has been slow due to the lack of lexical resources like pre-trained word embedding, language models and well-annotated datasets. Many people have put together labeled datasets for resource-constrained texts containing news articles, tweets, blog posts and other types of web texts. Few researchers have performed sentiment analysis on such datasets however, performing an emotion analysis is more complex task given the granularity of the labels. This section covers a variety of related works about the usage of machine learning and deep learning models for text-based emotion classification. Subsection 2.1 describes the related work for high-resourced languages like English, Chinese, Japanese, and some of European languages. Subsection 2.2 on the other hand outlines the current state of emotion analysis for low-resource languages.

2.1 Emotion analysis in high-resourced Languages

To classify emotions from text various NLP methods have been proposed - the keyword approach, the lexicon-based approach and the learning-based approach (Bharti et al. 2022). However, the first two approaches have a few limitations like lack of context as they totally depend on the presence of specific keywords from emotion lexicon, limited coverage of emotion lexicons leading to misclassification, poor performance for detecting some specific emotions and many more. On the other hand, in learning-based approach, various models are trained to give accurate and better results. The experiment was performed on three different datasets of varying sizes - ISEAR, WASSA, and Emotion-stimulus and showed that although deep learning models give better accuracy than machine learning models for large data but for small data, machine learning performs better. Madhu Midhan et al. (2023) investigated multiple machine-learning algorithms for analyzing mental health and classifying human emotions in texts. Various pre-processing techniques and their associated advantages and disadvantages has been described in the research. Texts are pre-processed using the Neat-Text NLP package which provides various inbuilt functions and text frames for text cleaning, stemming and tokenization. For feature extraction count vectorizer and TF-TDF approaches are used. TF-IDF evaluates a word's significance in a document while considering it's relationships to other terms in the same corpus. The highest accuracy achieved on the dataset was 62% by Logistic Regression model using count vectorizer suggesting the two feature extraction methods and the pre-processing technique can be helpful for emotion classification tasks. Nazarenko et al. (2021) analyzed various deep learning methods such as Word Embeddings, Bidirectional LSTM, Bidirectional Gated Recurrent Unit, Convolution Neural Network for emotion classification tasks. Deep learning models use word vectors to present text in n-dimensional space such that nearly identical/similar words are positioned very close together in the vector space. The experiment involved several types of word vector embeddings: GloVe, Word2Vec and FastText. In the case of datasets with balanced classes, BiLSTM and Word2Vec performed better whereas BiGRU with GloVe architecture showed better results for dataset with sparse classes. Results were evaluated on the basis of the F1 score, precision and recall of the model's output. The two studies are a good illustration of the basic steps for how text classification should be done for predicting emotions using learning-based approach. Suhasini & Srinivasu (2020) implemented machine learning approaches for Twitter messages for the detection of emotions. The study showed the efficiency of Naïve Bayes algorithm was higher when compared to the K Nearest Neighbour (KNN) and obtained an accuracy of 72.60%, 55.50% respectively. Xu et al. (2020) proposed a micro-blog emotion classification model, that uses CNN with Word2Vec word embedding for emotion classification from short/micro Chinese blogs. The word vectors are fed as input vectors to the model to learn text features. The overall accuracy achieved by CNN_Text_Word2vec was 7% higher than the ML models such as SVM, LSTM and RNN. Kannan & Kothamasu (2022) combined multiple Twitter dialogues spanning over five different emotion labels: joy, sad, anger, fear,and neutral and created a dataset for multi-class sentiment analysis with 13k sentences. Post-performing data pre-processing tasks like stemming, stop word removal and tokenization, TF-IDF was used for vector representation of the texts. Multiple traditional machine learning models like Naive Bayes, Random Forest, Logistic Regression were tested as the baseline models. However, the accuracy of the classification model increased by 24% with a fine tuned pre-trained BERT model. Simply layering one hidden-layer neural network classi-

fier on top of traditional BERT gave higher results with a minimal amount of data. For classifying emotions from Chinese short texts, Zhang et al. (2022) proposed a new method using ELECTRA and a hybrid neural network. Compared to BERT, ELECTRA model can avoid inconsistency of the mask training in the embedding layer of the model. A self-attention mechanism and a BiLSTM are selected to obtain a finer-grained representation of the review text in the training layer. In the output layer, the Softmax classifier classifies the input corpus according to the sentiment characteristics of the texts. In a few of the studies researchers also showed that hybrid models that combine two models can also exhibit good performance in NLP tasks. Li, Yi et al. (2022) used CNN-BiLSTM for emotion analysis from microblog comments, Gou et al. (2023) used BERT embeddings with BiLSTM for emotion analysis from two dialogue sets and observed the hybrid model outperformed baseline models significantly. For sentiment analysis in Chinese buzzwords, Li, Lei & Ji (2022) used BERT embeddings to capture rich context of the sentences and BiLSTM for feature extraction to obtain the local and global semantic features of the texts.

2.2 Emotion analysis in low-resourced languages

Although emotion detection from low-resourced languages is still in its infancy, few research works have been done on the topic focused on traditional machine learning and deep learning algorithms. Madhu Midhan et al. (2023) combined a comprehensive corpus of 18k Urdu sentences gathered from different domains with six different classes. Multiple baseline ML algorithms like KNN, Decision tree, SVM, and Random Forest were applied to this corpus. The input text was mapped to Word2Vec embedding for vector representations, which were further fed to the ML models. To counter the limitations of feature extraction techniques bag-of-words and n-gram models, Singh et al. (2019) proposed LSTM to assist humans in case of natural disaster by classifying tweets. Ranathunga & Liyanage (2021) conducted a comparative analysis with the use of state-of-the-art techniques such as Naive Bayes (NB), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Support Vector Machines (SVM), Long Short-term Memory (LSTM), and a CNN + SVM for multi-class sentiment analysis of Sinhala news comments. Although, due to the small size the accuracy difference between ML models and DL models was insignificant the experiment showed with word embeddings FastText and Word2Vec better performance can be achieved for resource-deprived languages like Sinhala.

Recently, transformer-based models have been introduced and are being progressively used for text classification tasks. Pre-trained models' training process does not begin from scratch, in contrast to DL approaches, which speed up the generalization of the model and the learning of the problem. In the study done by Ucan et al. (2022) pre-trained language models BERT and DistillBERT were used to classify emotions from Turkish texts. The feature engineering techniques like Bag Of Words and count vectorizer used in machine learning methods often cause loss of information as they don't consider the order and integrity of words in sentences. On the other hand, deep learning models preserve the context of the co-existing words and hence result in better classification results. However, the performance of deep learning models depends on the dataset size the larger a dataset is the better the model can learn the text features. In the study, a comparison between different ML, DL and transformer models was performed which suggested pre-trained language models are superior when it comes to low-resource languages like Turkish. Ozturk & Ozcan (2022) investigated the performance of transformer-based sen-

timent analysis models on four different Turkish datasets. After the data preprocessing phase and train-test split transformer-based BERT, ConvBERT, ELECTRA have been applied to classify sentences into three classes - positive, negative, or neutral, The results of the experiments showed that transformer-based models are superior to deep learning and traditional machine learning models in terms of F-score performance.
Das et al. (2021) developed a Bengali emotion corpus consisting of 6243 sentences from six different emotion classes: anger, disgust, fear, joy, sadness, and surprise. The performance of various ML, DNN, and transformer models was investigated in this corpus. Given their state-of-the-art performance in the classification task, three transformer models m-BERT, Bangla-BERT, and XLM-R on BEmoC were used for emotion classification from Bangla text. Out of ML models, LR gave the highest accuracy of 60.75% followed by 56.94% accuracy by BiLSTM deep learning model. However, the F1 and recall scores improved significantly, with the pre-trained models XLM-R giving the highest accuracy of 70.11%. Kannan et al. (2021) used IndicBERT, a mBERT model trained for Indian languages that captures semantic and linguistic features from multilingual texts. The approach was used for sentiment analysis in code-mixed Tamil tweets, and an F1 score of 61.73% was achieved.

3 Methodology

The research methodology entails six steps, namely, data gathering, exploratory data analysis, data pre-processing, data transformation, data modeling and conversion, evaluation, and result interpretation, as shown in Figure 1. The data mining methodology used is Knowledge Discovery in Databases (KDD).

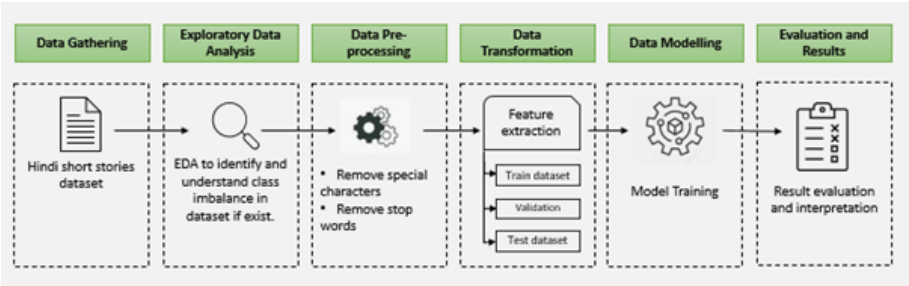


Figure 1: Research methodology

1. The *Data Gathering* step involves collecting input texts from a low-resourced language. The BHAAV (Kumar et al. 2019) dataset, used in the study, is an annotated corpus collected from 230 Hindi short stories. Each sentence has been classified with one of five emotions — angry, joyful, suspenseful, sad, and neutral. This dataset is publicly available and can be downloaded from zenodo.org [1] as an Excel file. Most recently, this dataset has been analyzed in only one study by (Kumar et al. 2023).

2. The *Exploratory Data Analysis* involves inspecting the structure and format of the dataset via plotting multiple data visualizations and checking data samples. BHAAV dataset has high-class imbalance where the 'Neutral' emotion class holds around 60% of the data. Data distribution among the five labels is shown in Figure

[1] `https://zenodo.org/record/3457467`

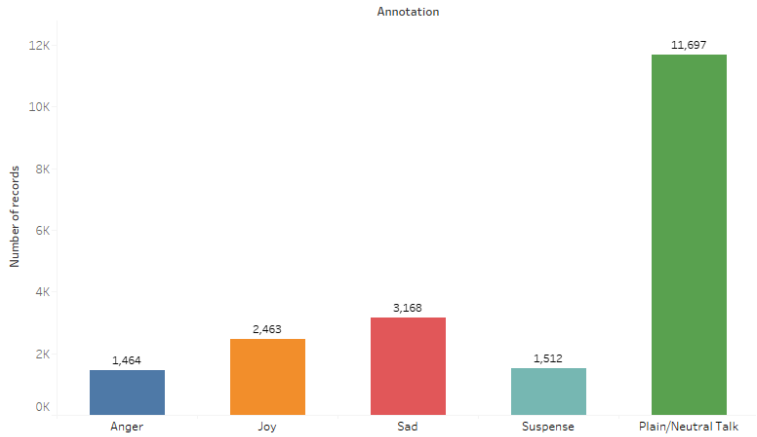


Figure 2: Dataset distribution for each emotion label

2. The class imbalance was removed before splitting the dataset for train, valid and test datasets.

3. The *Data Pre-processing* involves cleaning noise from the dataset. Removal of special characters like punctuation marks, numbers, and non-Hindi alphabets like," —;/][[0-9] was performed on the dataset. Stop words should also be removed from the sentences due to their low information content and high frequency in texts. Removing stop words from texts reduces feature space and increases focus on the essential features. Hindi stopwords for the study were downloaded from Kaggle[2].

4. The *Data Transformation* involves tokenizing the sentences into words and converting them to a numerical format that machine learning models can understand. For ML models, TF-IDF (Term Frequency-Inverse Document Frequency) was used for feature extraction and converting texts into vector space. For deep learning models pre-trained Fasttext and IndicFT, a Fasttext embedding model trained on Indian languages[3] word embedding model was used. By using word embedding, words are represented as real-valued vectors that encode their meaning, such that words with similar meanings are closer to each other in the vector space[4]. Before training the models for classification tasks, the input dataset is divided into three subsets: – Train dataset (80%), the test dataset (10%), and the validation dataset (10%).

5. *Data Modeling and Conclusion* involves building model architecture and model training. The models were trained with the split training dataset and tested with the validation dataset. Three baseline machine learning models—logistic regression, Support vector machine (SVM), and Random forests—were first trained for the classification task. CNN, BiLSTM, and CNN+BiLSTM deep learning models were trained for the emotion classification tasks. Two pre-trained models, BERT and IndicBERT were trained. A hybrid model, mBERT+BiLSTM was also created for the classification task. Transfer learning was employed to maximize feature extraction and selection. The transformer models were used from the Huggingface transformers[5] and fine-tuned on the corpus. The softmax activation function was

[2]https://www.kaggle.com/datasets/ruchi798/hindi-stopwords
[3]https://ai4bharat.iitm.ac.in/indicft
[4]https://en.wikipedia.org/wiki/Word_embedding
[5]https://huggingface.co/docs/transformers/indexlibrary

used in the final classification layer for the deep learning models as this is a multi-class classification problem. With the softmax function, the probabilities for each class in a multi-class classification can be estimated.

6. The final step of *Evaluation and Results* involves evaluating the performance of each of the machine learning and deep learning models. Since the dataset has multiple emotion class labels and there is a class imbalance, macro average precision, recall, and F1 score are the best metrics to evaluate models. In macro averaged values, every class is weighed uniformly; the scores are calculated independently for each class and then the average is taken thus treating each class equally. Equations (1), (2), and (3) are the formulas for precision, recall, and F1 score, respectively.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

4 Design & Implementation

This section specifies the overall flow of the study, the environmental setup, including the required Python libraries, the parameters used for the models, and the architectures of the models used. Figure 3 shows the overall architecture of the experiment.

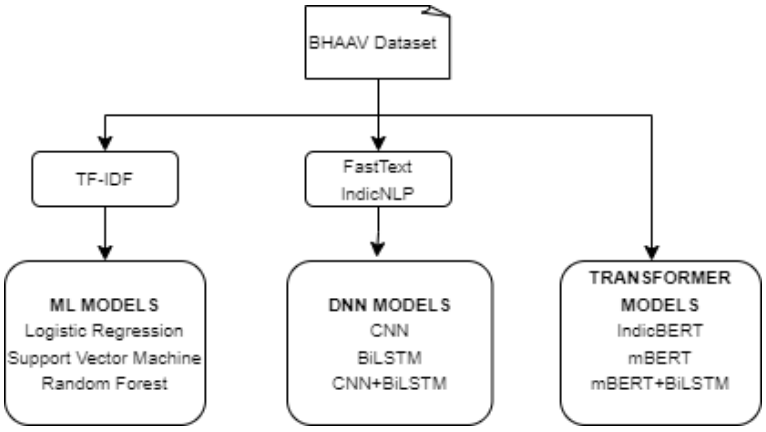


Figure 3: System architecture

For machine learning models, the term frequency-inverse document frequency (TF-IDF) was used to vectorize text data. Text is tokenized into unigrams and bigrams to capture the relationship between words, and their TF-IDF score is used as the feature value. FastText word embedding pre-trained on Hindi Wikipedia corpus[6] was used for text representation in deep learning models. For transformer models, pre-trained tokenizers like the mBERT tokenizer, and the IndicBERT tokenizers were used to tokenize the text. While tokenizing the text, input_id is created, which is a numerical representation of the text as mapped to the vocabulary of the pre-trained model.

[6] `https://fasttext.cc/docs/en/pretrained-vectors.html`

4.1 Environmental setup

The fine-tuning of pre-trained models requires lengthy processing times and intense computation power. In the research Deep Learning AMI GPU TensorFlow 2.13 (Ubuntu 20.04) 20230724 Amazon machine image (AMI) with p3.2xlarge configuration was used. Python was used for coding purposes, versions of all the libraries used in the experiments are shown in Table 1

Library	Version
python	3.10.12
transformers	4.31.0
tensorflow	2.13.0
pandas	2.0.3
openpyxl	3.1.2
indic-nlp-library	0.92
numpy	1.24.3
tensorflow	2.13.0
fasttext	0.9.2
scikit-learn	1.3.0

Table 1: Software specifications

4.2 Handling data class imbalance

To handle the class imbalance from the dataset SMOTE was used on the training dataset. SMOTE is an oversampling technique that handles class imbalance in datasets by generating synthetic samples for the minority classes. Table 2 below shows the emotion class distribution before and after SMOTE in BHAAV.

Class	Before SMOTE	After SMOTE
Angry	1176	9453
Joy	1996	9453
Sad	2584	9453
Suspense	1236	9453
Plain/Neutral	9453	9453

Table 2: Emotion class distribution in train dataset

It is important to apply SMOTE only on the training dataset and not on the validation or test datasets because doing so would introduce data leakage leading to model overfit on the test dataset and overly optimistic performance metrics and unreliable model evaluation results. In addition, another way of removing biases from an imbalanced dataset is to assign different class weights to both majority and minority labels in the dataset. The class weight for a label should be calculated based on its frequency in the dataset. scikit.learn provides a function compute_class_weight which estimates class weights for unbalanced datasets automatically[7].

[7] `https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html`

4.3 Model Implementation

In the experiment, various machine learning and deep learning models were investigated as baseline models. The hyperparameter tuning for the models was done using the GridSearchCV method with a cross-validation 'cv' value set to 5. Below is the detailed implementation description for the models used in the study:

- **Logistic Regression** Logistic regression model with solver 'newton-cg' and 'lbfgs' were used in the experiment since both of them are available for multi-class classification problems and work well in small to medium-sized datasets. The models were hyper-tuned for different values of regularization parameter C = [0.001, 0.01, 0.1, 1, 10, 100] and max_iter= [100, 250, 500, 1000, 10000] using sklearn GridSearchCV library using 10 fold cross validation technique. Train accuracy of 61% and 40% test accuracy was achieved.

- **Random Forest** Random forest classifier was hyper-tuned using GridSearchCV on parameters n_estimators = [50, 100, 150, 200, 250, 300, 400] and max_depth = [5, 10, 15, 20, 25, 50], min_samples_leaf= [1, 2, 4] and min_samples_split: [2, 5, 10]. Train accuracy of 57% and 41% test accuracy was achieved.

- **Support Vector machine** Support Vector machine was hyper-tuned using parameters regularization parameter c = [0.001, 0.01, 0.1, 1, 10, 100] and kernel= ['linear', 'poly', 'rbf'].

- **CNN** First, the data is tokenized using IndicNLP's 'indic_tokenize' and then the dataframe is used by Fasttext/IndicFT word embeddings to extract the features. For hyperparameter tuning manual grid search was used. After hyper-tuning the optimal parameters were learning rate, dropout = 0.4, epoch= 50 and a number of dense units =32 and 64 CNN filters with filter sizes = 4, activation function = 'ReLU', Dense layer activation = 'softmax' and optimizer function ='Adam' was used for the model. Since the problem is a multi-class classification sparse_categorical _crossentropy was used as the loss function.

- **BiLSTM** The BiLSTM created consists of a BiLSTM layer with 64 hidden units, and a fully connected layer having 32 neurons with 'ReLU' activation.

- **CNN+BiLSTM** Li, Yi et al. (2022) showed using a combination of CNN and BiLSTM for classification task can give better performance as it has the capability to capture previous and subsequent information in the text. On top of two BiLSTM layers with 64 and 32 units, a 1D convolutional layer with 64 filters of size three and a 1D max-pool layer were used. 'softmax' activation function was used in the final classification layer.

- **mBERT** Multilingual Bidirectional Encoder Representations from Transformers (mBERT) is pre-trained on a large corpus of 104 languages data in a self-supervised fashion. It's built with BERT base with 12 layers of transformers and 768 hidden layers. One method of fine-tuning a BERT model is by freezing-unfreezing different layers and evaluating the performance of the model (Morris & Street 2013) (Su & Vijay-Shanker 2022). Multiple combinations of freezing/unfreezing BERT layers were performed with different values of the learning rate, epoch, and batch_size.

- **IndicBERT** IndicBERT is a multilingual ALBERT model trained in 12 different Indian languages (Kakwani et al. 2020). The model can be downloaded from Huggingface transformer library. Different values of epoch = [10,15,20,20], batch_size = [15,20,25,35,40] and dropout =[0.3,0.4] were tested. With the increase in epoch and batch_size, model overfitting was observed. After multiple runs batch_size=20, epoch = 10, and dropout = 0.3 gave the highest accuracy. Train accuracy of 63% and test accuracy of 55% was achieved.

- **mBERT+BiLSTM** A hybrid model of mBERT and BiLSTM was created to experiment for the classification task. mBERT has the capability to capture contextual meanings from texts and it is trained on large dataset of multilingual texts like Hindi. In addition, BiLSTM ability to capture sequential patterns from the sentences allows it perform well on the text classification task. Figure 4 shows the model architecture of the hybrid model. On top mBERT layer was used to tokenize and capture embeddings from the texts and BilSTM layers were added later to capture the sequential patterns from the sentences.

```
>>> model.summary()
Model: "model_2"
_________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
=================================================================================================
 input_ids (InputLayer)      [(None, 227)]             0         []

 attention_mask (InputLayer  [(None, 227)]             0         []
 )

 tf_bert_model_2 (TFBertMod  TFBaseModelOutputWithPooli  1778534   ['input_ids[0][0]',
 el)                         ngAndCrossAttentions(last_  40         'attention_mask[0][0]']
                             hidden_state=(None, 227, 7
                             68),
                              pooler_output=(None, 768)
                             , past_key_values=None, hi
                             dden_states=None, attentio
                             ns=None, cross_attentions=
                             None)

 bidirectional_3 (Bidirecti  (None, 227, 128)          426496    ['tf_bert_model_2[0][0]']
 onal)

 dropout_117 (Dropout)       (None, 227, 128)          0         ['bidirectional_3[0][0]']

 dense_5 (Dense)             (None, 227, 64)           8256      ['dropout_117[0][0]']

 dropout_118 (Dropout)       (None, 227, 64)           0         ['dense_5[0][0]']

 dense_7 (Dense)             (None, 227, 5)            325       ['dropout_118[0][0]']

=================================================================================================
Total params: 178288517 (680.12 MB)
Trainable params: 178288517 (680.12 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________________________________________
>>>
```

Figure 4: mbert+BiLSTM model summary

5 Evaluation

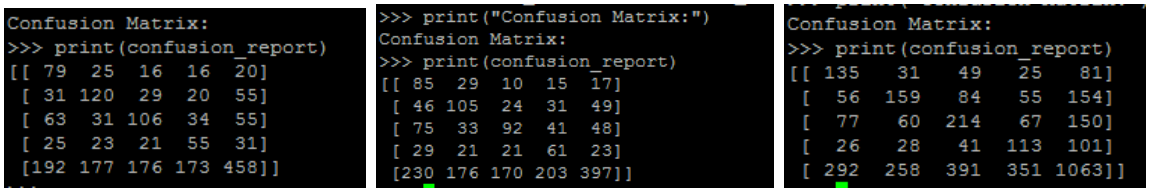
In this section the evaluation metrics shown have been calculated for the test dataset. A model's performance can be best evaluated based on it's performance on unseen data.

5.1 Experiment 1: Machine Learning Models

The first experiment includes implementing baseline machine learning models. Table 3 shows the macro average F1 score, macro average precision, macro average recall, and balanced accuracy for the ML models and figure 5 shows the corresponding confusion matrix.

Model	Macro Avg Precision	Macro Avg Recall	Macro Avg F1	Accuracy
Logistic Regression	0.35	0.42	0.35	0.40
Random Forest	0.33	0.39	0.34	0.41
SVM	0.36	0.40	0.32	0.36

Table 3: Performance of baseline machine learning models on the BHAAV dataset



(a) Logistic Regression (b) Support Vector Machine (c) Random Forest

Figure 5: Confusion matrix for ML models.

5.2 Experiment 2: Deep Learning Models

In the second experiment baseline deep learning models were implemented. Table 4 shows the macro average F1 score, macro average precision, macro average recall, and balanced accuracy the DNN models.

Model	Word Embedding	Macro Avg Precision	Macro Avg Recall	Macro Avg F1	Accuracy
CNN	FastText	0.30	0.37	0.29	0.31
CNN	IndicFT	0.35	0.43	0.35	0.38
BiLSTM	FastText	0.35	0.43	0.34	0.38
BiLSTM	IndicFT	0.37	0.45	0.36	0.38
CNN+BiLSTM	FastText	0.35	0.43	0.34	0.38
CNN+BiLSTM	IndicFT	0.35	0.45	0.34	0.34

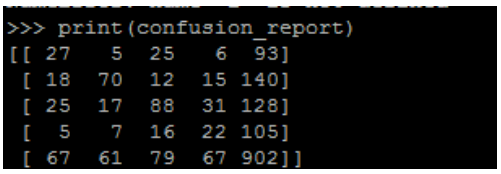
Table 4: Performance of baseline deep learning models on the BHAAV dataset

5.3 Experiment 3: Transformer Models

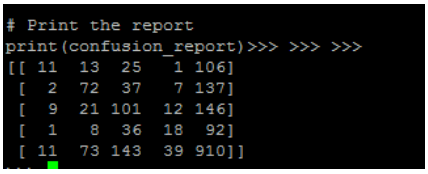
The third experiment includes fin tuning and implementing pre-trained transformer models. Table 5 shows the macro average F1 score, macro average precision, macro average recall, and balanced accuracy of the transformer models and figure 6 shows the corresponding confusion matrix.

Model	Macro Avg Precision	Macro Avg Recall	Macro Avg F1	Accuracy
mBERT	0.38	0.32	0.33	0.55
indicBERT	0.37	0.33	0.34	0.55
mBERT+BiLSTM	0.41	0.40	0.39	0.57

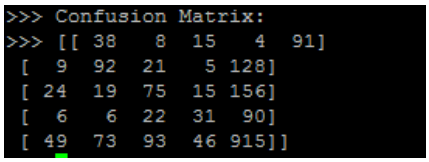
Table 5: Performance of baseline machine learning models on the BHAAV dataset



(a) IndicBERT



(b) mBERT



(c) mBERT+BiLSTM

Figure 6: Confusion matrix for Transformer models.

5.4 Discussion

On comparing the evaluation metrics of ML models it was observed the models are almost similar in performance. Although, Random Forest has the highest test accuracy of 41% macro average precision and recall are still low. Although class weights were applied during the model training but from the confusion matrix, it was observed the model is able to classify the majority class i.e. 'Neutral/Plain Talk' more accurately than the other minority classes, and hence the low precision and recall values. On the other hand, the performance of deep baseline models was very low compared to the ML models. BiLSTM with both FastText and IndicFT gave the highest test accuracy of 38%. The deep models have a macro average 43% recall which shows the models are correctly predicting a good percentage of actual positive instances. The models trained by Kumar et al. (2019) in the base paper achieved high test accuracy between 50-60% for all the models, this difference might be the reason for different sets of hyperparameters used in the BHAAV research. In deep models, BiLSTM performed well with both FastText and IndicFT embedding achieving a test accuracy of 38% and macro average recall of 45%. Although, mBERT outperformed baseline models with an accuracy of 55% but the macro average recall and precision value were very low suggesting that the model is performing well on majority labels but struggling on minority labels. Finally, a hybrid model mBERT+BiLSTM was created to get the best of both the good-performing models. With the hybrid model test accuracy improved significantly reaching 57%. Furthermore, the macro average metrics also improved slightly showing the hybrid was able to perform better with both the majority and minority labels.

6 Conclusion and Future Work

The aim of this research was to verify the extent to which deep learning models can be leveraged to capture emotions from low resourced texts. This research proposes a deep learning framework that uses mBERT and BiLSTM to capture the contextual meaning from Hindi texts. Results from the comparative study between ML and DL models demonstrate that pre-trained models outperform baseline models. A limitation of this study was scarcity of annotated Hindi corpus to train the models more and capture diverse emotions. The expectation of this study was that the pre-trained models would

perform exceptionally well on the low resourced language given their history to work well with high resourced languages. However, same couldn't be achieved in this study. With fine tuning the pre-trained models more, experimenting with more class imbalance removal methods and using data augmentation to create more samples for models to train hopefully the expectations can be achieved. This research can potentially enhance the method of analyzing sentiments behind low resourced texts which in turn can be helpful for businesses, government and associated entities to take informed decisions. Furthermore, an extensive research can be carried out in this study by creating and capturing more annotated datasets. Effective data augmentation techniques like back translation, synonym word translation and many more can be used to increase the training samples for the models. With recent progress in pre-trained generative models like ChatGPT -3 generating emotion specific data in any low resourced languages can be achieved. These models are proven to produce data with comparatively higher accuracy.

References

Acheampong, F. A., Wenyu, C. & Nunoo-Mensah, H. (2020), 'Text-based emotion detection: Advances, challenges, and opportunities', *Engineering Reports* **2**(7), e12189.

Alam, T., Khan, A. & Alam, F. (2020), 'Bangla text classification using transformers', *arXiv preprint arXiv:2011.04446* .

Bharti, S. K., Varadhaganapathy, S., Gupta, R. K., Shukla, P. K., Bouye, M., Hingaa, S. K. & Mahmoud, A. (2022), 'Text-based emotion recognition using deep learning approach', *Computational Intelligence and Neuroscience* **2022**.

Das, A., Sharif, O., Hoque, M. M. & Sarker, I. H. (2021), 'Emotion classification in a resource constrained language using transformer-based approach', *arXiv preprint arXiv:2104.08613* .

Gou, Z., Li, Y. et al. (2023), 'Integrating bert embeddings and bilstm for emotion analysis of dialogue', *Computational Intelligence and Neuroscience* **2023**.

Kakwani, D., Kunchukuttan, A., Golla, S., N.C., G., Bhattacharyya, A., Khapra, M. M. & Kumar, P. (2020), IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages, *in* 'Findings of EMNLP'.

Kannan, E. & Kothamasu, L. A. (2022), 'Fine-tuning bert based approach for multi-class sentiment analysis on twitter emotion data.', *Ingénierie des Systèmes d'Information* **27**(1).

Kannan, R. R., Rajalakshmi, R. & Kumar, L. (2021), 'Indicbert based approach for sentiment analysis on code-mixed tamil tweets'.

Kumar, T., Mahrishi, M. & Sharma, G. (2023), 'Emotion recognition in hindi text using multilingual bert transformer', *Multimedia Tools and Applications* pp. 1–22.

Kumar, Y., Mahata, D., Aggarwal, S., Chugh, A., Maheshwari, R. & Shah, R. R. (2019), 'Bhaav-a text corpus for emotion analysis from hindi stories', *arXiv preprint arXiv:1910.04073* .

Li, A., Yi, S. et al. (2022), 'Emotion analysis model of microblog comment text based on cnn-bilstm', *Computational Intelligence and Neuroscience* **2022**.

Li, X., Lei, Y. & Ji, S. (2022), 'Bert-and bilstm-based sentiment analysis of online chinese buzzwords', *Future Internet* **14**(11), 332.

Madhu Midhan, T., Selvaraj, P., Harshavardan Kumar Raju., M., Bhanu Prakash Reddy., M. & Bhaskar., T. (2023), Classification of mental health and emotion of human from text using machine learning approaches, *in* '2023 6th International Conference on Information Systems and Computer Networks (ISCON)', pp. 1–7.

Magueresse, A., Carles, V. & Heetderks, E. (2020), 'Low-resource languages: A review of past work and future challenges', *arXiv preprint arXiv:2006.07264* .

Morris, C. B. J. & Street, R. (2013), 'Brain systems mediating aversive conditioning: an event-related fmrl study', *Fear and Anxiety: The Science of Mental Health* **10**, 199.

Nazarenko, D., Afanasieva, I., Golian, N. & Golian, V. (2021), Investigation of the deep learning approaches to classify emotions in texts., *in* 'COLINS', pp. 206–224.

Ozturk, O. & Ozcan, A. (2022), Sentiment analysis in turkish using transformer-based deep learning models, *in* 'The International Conference on Artificial Intelligence and Applied Mathematics in Engineering', Springer, pp. 1–15.

Ranathunga, S. & Liyanage, I. U. (2021), 'Sentiment analysis of sinhala news comments', *Transactions on Asian and Low-Resource Language Information Processing* **20**(4), 1–23.

Singh, J. P., Dwivedi, Y. K., Rana, N. P., Kumar, A. & Kapoor, K. K. (2019), 'Event classification and location prediction from tweets during disasters', *Annals of Operations Research* **283**, 737–757.

Su, P. & Vijay-Shanker, K. (2022), 'Investigation of improving the pre-training and fine-tuning of bert model for biomedical relation extraction', *BMC bioinformatics* **23**(1), 120.

Suhasini, M. & Srinivasu, B. (2020), Emotion detection framework for twitter data using supervised classifiers, *in* 'Data Engineering and Communication Technology: Proceedings of 3rd ICDECT-2K19', Springer, pp. 565–576.

Ucan, A., Dörterler, M. & Akçapınar Sezer, E. (2022), 'A study of turkish emotion classification with pretrained language models', *Journal of Information Science* **48**(6), 857–865.

Xu, D., Tian, Z., Lai, R., Kong, X., Tan, Z. & Shi, W. (2020), 'Deep learning based emotion analysis of microblog texts', *Information Fusion* **64**, 1–11.

Zhang, S., Yu, H. & Zhu, G. (2022), 'An emotional classification method of chinese short comment text based on electra', *Connection Science* **34**(1), 254–273.