SN Computer Science (2022) 3:62
https://doi.org/10.1007/s42979-021-00913-0

ORIGINAL RESEARCH

Multi-class Textual Emotion Categorization using Ensemble of Convolutional and Recurrent Neural Network

Tanzia Parvin[1] · **Omar Sharif**[1] · **Mohammed Moshiul Hoque**[1]

Received: 27 August 2021 / Accepted: 30 September 2021 / Published online: 11 November 2021

Abstract

Categorizing emotion refers to extracting the individuals' behaviour from texts and assigning textual units into an emotion from predefined emotional connotations. Identification and categorization of emotion content have mostly been made for English, French, Chinese, Arabic, and other high-resource languages. However, very few studies have investigated emotion from the under-resourced language like Bengali. This work proposes an ensemble-based technique for classifying textual emotions into six classes: anger, disgust, fear, joy, sadness and surprise. An emotion corpus containing 9000 Bengali texts is developed to perform the emotion classification. This work investigates 22 standard classifier models developing based on three deep learning techniques (Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), Bidirectional Long Short Term Memory (BiLSTM) with different ensemble strategies and embedding models (i.e., Word2Vec, FastText). All the models are tuned, trained and tested on the developed dataset (EBEmoD-Extended Bengali Emotion Dataset) and a publicly available emotion dataset (BYCD-Bengali Youtube Comment dataset). The experimental result demonstrates that the ensemble of CNN and BiLSTM (i.e., CNN+BiLSTM) outdoes all other models by acquiring the highest weighted f_1 -score of 62.46% (for EBEmoD) and 67.57% (for BYCD), respectively.

Keywords Natural language processing · Textual emotion classification · Deep learning · Emotion corpus · Ensemble technique

Introduction

The extraordinary advancement of the internet and social media platforms enables individuals to express their opinions, feelings, or emotions. A large portion of social media users interact with each other and share their emotions, experiences and opinions through tweets, reviews, messaging posts, and comments in the form of text. This textual data reflects the emotional propensities of the users towards different aspects such as politics, business, mental health, and societal issues [27]. The availability of enormous textual data has accelerated the interest in emotion analysis research. Automatic emotion categorization is assigning textual units into an emotion from a set of predefined classes. Usually, it refers to extracting the psychology of individuals' behaviour from texts that reveal different emotional connotations such as joy, sadness, surprise, disgust, anger, etc. Although a significant amount of researches have been carried out regarding textual emotion categorization for high resource languages like English, French, Chinese, Arabic [2, 11]. However, very few researchers have investigated this issue in under-resourced languages like Bengali. With around 265 million native speakers, Bengali is the 7th most spoken language [25]. A massive number of people communicate through virtual platforms using territorial languages (i.e., Bengali). Thus, developing various NLP tools or automated systems for Bengali is a growing demand that can provide insights into an individual's feelings about an event,

This article is part of the topical collection "Enabling Innovative Computational Intelligence Technologies for IOT" guest edited by Omer Rana, Rajiv Misra, Alexander Pfeiffer, Luigi Troiano and Nishtha Kesswani.

✉ Mohammed Moshiul Hoque
moshiul_240@cuet.ac.bd

Tanzia Parvin
tanzia.mim1376@gmail.com

Omar Sharif
omar.sharif@cuet.ac.bd

[1] Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong 4349, Bangladesh

product, service, etc. Such a system can improve the product or service quality, change sales strategies, and predict future trends. Moreover, an automatic emotion analysis system can shape brand image, track customer response, identify cases of cyber-bullying, detect public sentiment, track well-being.

However, emotion classification from Bengali text is quite challenging due to the deficit of benchmark corpora, complex morphological structure, critical linguistic construct, and huge verb inflexions. Deep learning models have recently shown significant improvements to classify textual emotion [1, 8]. Therefore, this work aims to apply deep learning methods to categorize Bengali texts into one of six basic emotion (e.g., anger, disgust, fear, joy, sadness, surprise) classes defined by Ekman [10]. Many works in other languages have already adopted Ekman's taxonomy to recognize textual emotion [24]. To attain the goal, a corpus of 9000 Bengali texts is developed, considering six emotion classes. Subsequently, a set of classifier models is developed using deep learning techniques (i.e., CNN, GRU, BiLSTM) with different ensemble strategies and embedding models (e.g. Word2vec, FastText). An extensive experiment is performed on the developed dataset, and a publicly available dataset [29]. The key contributions can be illustrated as follows:

- Develop a dataset of 9000 Bengali texts considering six (i.e., anger, disgust, fear, joy, sadness, surprise) emotion categories.
- Develop a weighted ensemble model using CNN and BiLSTM to classify textual emotion. The proposed model outdoes other machine and deep learning baselines by achieving the highest f_1-score.
- Empirically evaluates models performance on two different datasets and demonstrate how the proposed ensemble model can increase the model's predictive accuracy.

Related Work

Identification and analysis of textual emotion contents have attracted much attention from researchers in the last couple of years. Many works on emotion classification have been conducted in many languages such as English, Chinese and Arabic [2]. Balli et al. [4] applied Convolutional Neural Network(CNN) to categorize four basic emotions (such as happiness, anger, fear, sadness) from the Arabic tweets. They performed a comparative analysis with Support Vector Machine (SVM), Naive Bayes (NB) and Multi-layer Perceptron (MLP). Haryadi et al. [12] used LSTM, Nested LSTM and SVM to detect emotion from English tweets, and their Nested LSTM method achieved the best accuracy of 99.167%. A syntax-based graph convolution network (GCN) model is presented by Lia et al. [15] to classify emotion from Chinese microblogs, and their model achieved a f_1-score of 82.32%. Abdullah et al. [1] developed a CNN-LSTM based model to detect sentiments and emotions from Arabic texts. Alzu'bi et al. [3] offered a multi-label multi-target dataset of 11503 tweets with six emotion categories. Baseline evaluation is performed using DT, RF and KNN. Among the three models, RF performed better with the highest f_1 score of 82.6%. Mamta et al. [18] developed a multi-domain corpus of 12,737 English tweets for sentiment analysis. They employed a deep learning-based ensemble technique with CNN, LSTM, and GRU and obtained a weighted f_1 score of 84.7%.

To the best of our knowledge, a limited number of researches have been carried out to classify emotion from Bengali texts. Lora et al. [16] used different deep learning models (stacked LSTM, stacked LSTM with 1D convolution, CNN and RNN) to identify positive and negative emotions from Bengali texts. The RNN model with Glove embedding outperformed others by achieving 98.3% accuracy. Tripto et al. [29] applied LSTM and CNN to classify sentiment and emotion of romanized Bengali texts. The system achieved 54.24% and 59.23% accuracy in sentiment and emotion classification tasks. Another work was conducted by Rayhan et al. [23] to predict six emotions (happy, sad, fear, anger, love, surprise) from 7214 Bengali texts. They implemented two models: BiGRU and combined CNN-BiLSTM. The CNN-BiLSTM model outperformed BiGRU with an overall accuracy of 66.62%. A corpus of 2492 texts was developed by Rahman et al. [21] to classify Bengali sports news comments into five categories (e.g., happiness, sadness, advice, annoyance, neutral). They conducted various deep learning models like CNN, Multilayer Perceptron and LSTM, where the CNN model achieved the highest f_1 score of 48.19%. Pal et al. [20] adopted LR, KNN, SVM with linear kernel, RF and CNN to categorize four emotion classes (e.g., joy, anger, sadness, suspense) and achieved the highest accuracy of 73% using LR.

Datasets

All classification models utilized two datasets for training, validation, and testing.

- *Bengali Youtube Comment Dataset (BYCD)* It is the YouTube comment emotion dataset offered by Tripto et al. [29]. This dataset consists of YouTube comments in Bengali, English and romanized Bengali. Only the Bengali part of the dataset has been used to diminish the experimental complexity.
- *Extended Bengali Emotion Dataset (EBEmoD)* A Bengali emotion dataset contains six emotion classes (anger, disgust, fear, joy, sadness, surprise). We followed the

standard steps (crawling, preprocessing, annotation, label verification) to create an emotion annotated dataset described by Das et al. [7]. Ekman's [10] definition of emotion classes has been utilized to ensure the consistency of the samples in the dataset. In the previous work [7], the authors did not develop any model to classify emotion. This work extended their work by presenting an automatic emotion categorization system trained over a dataset of 9000 samples. Cohen's kappa score [6] is measured to ensure the quality of annotation. A kappa score of 82.43% is obtained, indicating substantial agreement between the annotators.

For training and evaluation, both the datasets are partitioned into train, validation, and test set. Table 1 illustrates a summary of the datasets. EBEmoD comprises 9000 text samples with six emotion classes, while BYCD comprises 753 texts from four (anger, disgust, joy, surprise) emotion classes. The disgust class has the highest number of instances on EBEmoD (2080 texts) and BYCD (305 texts). Moreover, the anger class contained 1171 texts in EBEmoD, and the surprise class consisted of the least samples (56 texts) in BYCD.

For better understandings, the training set is further analyzed concerning different attributes. Statistics of the training set are manifested in Table 2. It is observed that the *disgust* class contained a higher number of words and unique words on both datasets. In BYCD, the *disgust* class has seven times as many as total words compare to the surprise class. On average, the *sad* class in EBEmoD contains approximately 23 unique words per text. The *disgust* class in BYCD has approximately 16 words per text, while the joy class is only nine words long. Instances of BYCD are shorter in length than EBEmoD. On average, EBEmoD has more than 20 words in each text, while BYCD has less than 15 words.

System Overview

The primary concern of this research is to develop an automatic system that can categorize Bengali texts into six predefined emotion classes. Figure 1 exhibits the abstract process of the emotion categorization system, which consists of four main modules: preprocessing, feature extraction, classification model generation and prediction.

Preprocessing

Raw data can have errors, duplication, noises and superfluous information. Preprocessing is required to remove the inconsistencies as it helps to achieve accurate analytical results. Automated preprocessing is performed by removing punctuation, digits and unwanted characters from texts. A list (U) of unnecessary words, characters and punctuation is created, and texts are converted into a set of words

Table 1 Summery of the train, validation and test set of EBEmoD and BYCD

Class	EBEmoD			BYCD		
	Train	Val	Test	Train	Val	Test
Anger	924	140	107	167	19	16
Disgust	1609	185	286	246	26	33
Fear	1117	151	142	–	–	–
Joy	1335	167	151	144	22	24
Sadness	1148	125	135	–	–	–
Surprise	1067	132	139	45	8	3
	7200	900	900	602	75	76

Table 2 Training set statistics

	Dataset	Anger	Disgust	Fear	Joy	Sadness	Surprise
Total words	EBEmoD	20,092	34,531	22,300	26,693	26,965	22,581
	BYCD	2434	3816	–	1354	–	551
Unique words	EBEmoD	7001	8713	6255	8333	7977	6964
	BYCD	1267	1781	–	691	–	380
Max. text (in words)	EBEmoD	100	99	73	216	115	72
	BYCD	160	90	–	108	–	43
Avg. no. of words (per text)	EBEmoD	21.74	21.46	19.96	19.98	23.43	21.16
	BYCD	14.57	15.51	–	9.40	–	12.24

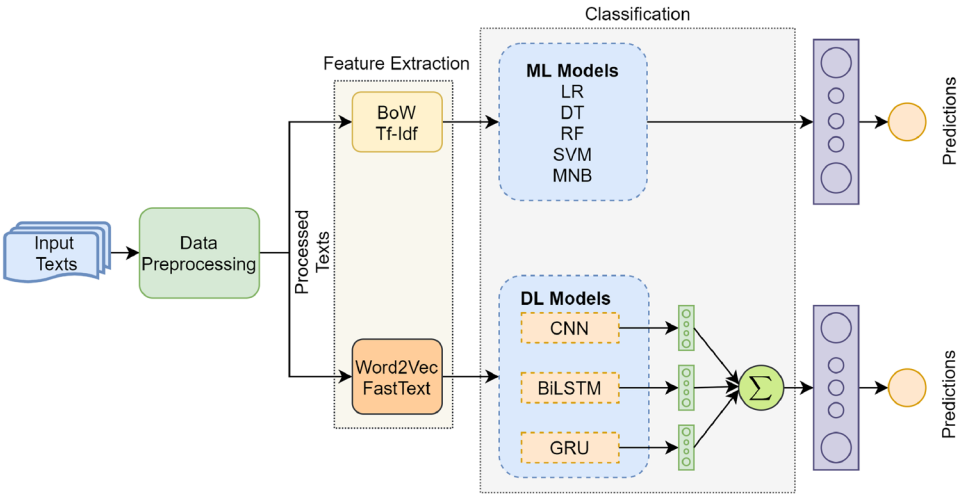

Fig. 1 Abstract process of the emotion classification system

using the tokenizer method. Finally, unnecessary tokens are discarded after matching with U.

Feature Extraction

Feature extraction technique transforming text into a numerical representation in a vector form for training the classifiers. This work investigates a few most widely used textual feature extraction techniques such as Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and word embeddings (i.e., Word2Vec, FastText).

- *BoW* the number of occurrences of each word within a document is used as features extracted [30]. A vocabulary of 15k feature words is created considering the unique words in the corpus.
- *TF-IDF* is used to reduce the impact of less informative words that appear very frequently in the texts [28]. The term frequency (TF) is the occurrence of a particular word in a document, and inverse document frequency (IDF) measures the significance of a word in the whole corpus. The 15k most frequent words are considered to extract the combination of unigram and bigram features.
- *Word Embedding* Word2Vec [19] and FastText [5] embedding techniques are applied to extract the semantic features of the texts. A vocabulary of 26000 unique words is constructed to get the features, and each sentence is mapped into a variable-length sequence vector based on the word index in vocabulary. Then each sequence is converted into a fixed-length vector of size 100 using the Keras pad sequence method. Finally, Word2Vec and FastText techniques are employed to learn the features where the embedding dimension is settled to 300.

Classification Approaches

Several popular machine learning (ML) and deep learning (DL) classifiers are explored for the textual emotion categorization task. The classifier generates a *classifier model* from the training feature vectors. The feature extractor transforms unseen text to feature sets and fed into the *classifier model* to predict emotion categories or classes (e.g., joy, anger).

ML Classifiers

Five well-known ML algorithms, such as Logistic regression (LR), Decision tree (DT), Random Forest (RF), Multinomial naive Bayes (MNB), and Support vector machine (SVM), are implemented for investigating the performance of textual emotion categorization task. Various combinations of parameters are tested and tuned to train the classifier models. The 'lbfgs' optimizer with 'l2' regularization is used to train the LR model for 300 iterations where the value of C is fixed to 1.0. For RF, 80 decision trees with 15k maximum features have been considered. The 'gini' and 'entropy' criterion is used to measure the quality of a split concerning RF and DT. We implemented SVM with an 'rbf' kernel where the tolerance and random state are set to 0.002 and 40.

DL Classifiers

To develop the classifier models, the three most popular deep learning architectures are considered as the base classifiers such as Convolution Neural Network (CNN), Bidirectional Long Short Term Memory (BiLSTM), and Gated Recurrent Unit (GRU). Eight ensemble-based classifiers (4 weighted ensembles, 4 average ensembles) were also developed using the combination of base classifiers.

- *CNN* popularly used in various text classification tasks due to their capability of capturing syntactic and semantic features of the texts [14, 26]. This work considers a CNN with two convolution layers where the first layer comprises 128 filters with kernel size 5 and the second layer contains 64 filters with kernel size 3. To downsample the features max-pooling technique is applied with window sizes 5 and 3. Finally, the 'relu' activation function is employed to add non-linearity and output of the 'softmax' layer used as the prediction.
- *BiLSTM* extracts contextual information from feature sequences by considering dependencies from both past and future [13]. We employed a BiLSTM model that consists of two layers similar to the CNN model with 128 and 64 cells, respectively. The dropout technique is used with a dropout rate of 0.2 to avoid overfitting, and the softmax layer is used for the final prediction.
- *GRU* captures the sequence information of various time scales from large sequences of data [17]. GRU is simpler than LSTM and takes less time to train than LSTM due to the smaller number of trainable parameters. Similar to BiLSTM, two layers of GRU having 128 and 64 recurrent units is utilized for model building.
- *Ensemble* This technique combines base classifiers to develop a specific predictive model while exploiting the individual classifier's strength. This work employs two ensemble techniques: average ensemble (AE) and weighted ensemble (WE). In AE, softmax probabilities of the participating models are averaged, and the class with the highest probability is considered output. In this technique, the prior performance of the base models is not taken into account during classification. All the classifier models are given similar priority. Rather than traditional averaging, the WE offers additional weights to the base classifiers softmax outcome based on heuristics and prior results [9]. This work considered the f_1-score of the classifiers on the validation set as the heuristic weight of the ensemble. Figure 2 depicts an overview of the proposed weighted-ensemble model for textual emotion classification.

Let consider for each instances, every participating model M_i of the ensemble provide softmax probability vector $P[]$. Thus for n participating models outputs are $P_1[], P_2[], ..., P_n[]$. Prior f_1-scores of the models on validation set are $F_1[], F_2[], ..., F_n[]$. Utilizing this values output of the weighted ensemble can be computed by Eq. 1,

$$O = \arg\max \left(\frac{\sum_{i=1}^{n} P_i[] * F[]}{\sum_{i=1}^{n} F[]} \right) \tag{1}$$

Here, O denotes the output predictions of the ensemble model. The process of calculating ensemble weights of the proposed model is presented in algorithm 1. After multiplying with the f_1-scores, softmax probabilities are aggregated and normalization is done by diving this with the sum of f_1-scores.

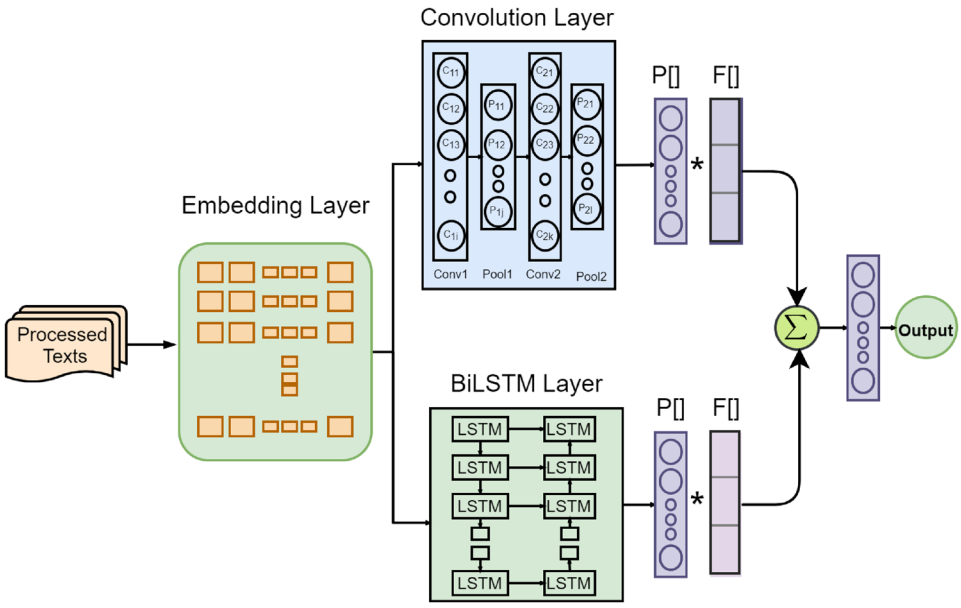

Fig. 2 Overview of the proposed weighted-ensemble model

Algorithm 1: Process of calculating ensemble weights

```
Input: Softmax probabilities and F1-score
Output: Predictions
P ← [] (softmax probabilities);
F ← [] (weighted f1 scores);
W ← [] (weighted sum);
for iϵ(1, n) do
    W[i] = W[i] + (Pi ∗ Fi);
    i = i + 1;
end
sum = 0;
for iϵ(1, n) do
    sum = sum + Fi;
    i = i + 1;
end
X = (W/sum) //normalized probabilities;
O = arg max(X) // set of predictions;
```

Since hyperparameter combination directly impacts the model's outcome, models are trained with different combinations. Optimal hyperparameter combination is selected based on models performance on the validation set by trial and error approach. A detailed summary of the hyperparameter values employed in DL models is illustrated in Table 3. All the models used 'adam' optimizer with a learning rate of 0.001. Moreover, models are trained for 20 epochs with 32 instances per batch. Based on the performance of the validation set, the best model is stored using the callback. A similar architecture is utilized on both datasets.

Experiments

This section provides a detailed performance analysis of different ML, DL, and ensemble-based models to categorize Bengali textual emotion on different datasets.

Experimental Details

For experimenting, we used the google colaboratory platform with the python 3.6.9 package. For the task of data preprocessing, pandas 1.1.4 and numpy 1.18.5 are used. All the models in ML are implemented with scikit-learn 0.22.2, while Keras 2.4.0 and TensorFlow 2.3.0 are utilized for training DL models. This work utilizes a weighted f_1 -score to determine the superiority of the models. However, precision and recall are also reported to compare the performances. Moreover, a detailed error analysis of the proposed weighted ensemble method is also presented in this section.

Results

Table 4 exhibits the performance comparison between developed models for EBEmoD. Among ML models, LR with BoW features achieved the highest f_1-score of 56.48%. MNB and SVM with BoW features also attained good outcomes of 55.75% and 55.81%, respectively. In contrast, with TF-IDF

Table 3 Hyperparameter summary of DL models

Hyperparameters	Hyperparameter space	Optimal Value
Embedding dimension	50 ,100, 150, 200, 300, 400	300
Filters (CNN)	12, 16, 32, 64 128, 256, 512	128, 64
Pooling types	'max', 'average'	'max'
Kernel size	3, 5, 7, 9	5, 3
Recurrent units (BiLSTM, GRU)	32, 64 128, 256	128, 64
Batch size	16, 32, 64	32
Dropout	0.1, 0.15, 0.20, 0.25, 0.30, 0.35	0.20
Optimizer	'adam' 'Nadam', 'RMSprop',	'adam'
Learning rate	0.3, 0.2, 0.1, 0.001, 0.0001, 0.0005	0.001
No of epochs	–	20

Table 4 Performance comparison of various ML and DL methods on the test set for different feature extraction (FE) techniques on EBEmoD

FE technique		Methods	Precision (%)	Recall (%)	F_1-score (%)
BoW	–	LR	56.69	56.44	56.48
		DT	44.85	43.67	44.15
		RF	53.87	53.11	52.31
		MNB	56.32	55.98	55.75
		SVM	58.37	55.11	55.81
TF-IDF	–	LR	58.10	57.33	57.49
		DT	45.19	44.33	44.63
		RF	58.03	54.89	53.90
		MNB	64.87	58.89	55.99
		SVM	60.06	58.05	57.85
Word2Vec	Base	CNN (C)	60.94	59.56	59.96
		BiLSTM (B)	57.57	54.11	55.24
		GRU (G)	60.05	57.67	57.71
	AE	C + B	59.66	56.67	57.71
		C + G	61.25	59.11	59.78
		B + G	60.38	58.22	58.87
		C + B + G	63.43	60.89	61.65
	WE	C + B	59.85	56.89	57.90
		C + G	61.38	59.22	59.89
		B + G	60.49	58.22	58.94
		C + B + G	63.01	60.55	61.30
FastText	Base	CNN (C)	58.31	59.67	58.64
		BiLSTM (B)	62.52	63.11	62.22
		GRU (G)	57.39	58.89	56.47
	AE	C + B	62.22	63.63	62.27
		C + G	59.23	60.55	58.82
		B + G	62.57	63.22	62.11
		C + B + G	62.62	63.56	62.17
	WE	C + B	62.48	63.44	62.46
		C + G	59.16	60.55	58.84
		B + G	62.67	63.33	62.21
		C + B + G	62.31	63.33	62.01

AE, WE indicates average and weighted ensemble respectively

features SVM gained maximal f_1-score of 57.85% amid all ML models. In the case of DL models, BiLSTM with FastText embedding outstripped others by achieving f_1-score of 62.22%. The obtained result is about 4% higher than the best ML model outcomes (SVM with TF-IDF). Initially, 16 different models (10 ML, 6 DL) are investigated in terms of precision, recall, and f_1-score. Out of 16, three best-performing methods (i.e., CNN, BiLSTM, GRU) are selected for the ensemble. Average and weighted ensemble techniques are applied to all possible combinations of these three base models. Results indicate that the weighted ensemble method with CNN and BiLSTM (i.e., C + B) obtained the highest f_1-score of 62.46% for FastText embedding, outperforming all other models.

Evaluation results of various classifiers on BYCD are illustrated in Table 5. A significant reduction in the system performance was observed with ML models on BYCD. The LR model with TF-IDF acquired the highest f_1-score of 47.13% which is around 10% less than the best ML model (SVM with TF-IDF) outcomes of EBEmoD. However, in BYCD, the DL models performed better with both the embeddings than EBEmoD. Like EBEmoD, BiLSTM with FastText obtained the maximum f_1-score amid DL models, which is 66.78%. It is not surprising that the proposed weighted ensemble method (i.e., C + B) has proven superior by achieving the highest f_1-score of 67.57% for BYCD. Thus, it is confirmed that the proposed weighted ensemble method (i.e., C + B) with FastText embedding outperformed all other models in both EBEmoD and BYCD. The individual strength of these models might be the reason behind achieving the improved performance. Convolution layers can extract the texts' prominent features, and BiLSTM captures

Table 5 Performance comparison of various ML and DL models on the test set for different feature extraction (FE) techniques on BYCD

FE technique		Methods	Precision (%)	Recall (%)	F_1-score (%)
BoW	–	LR	43.65	49.33	46.14
		DT	38.17	37.33	37.25
		RF	42.97	44.00	40.43
		MNB	45.70	48.00	46.17
		SVM	45.02	46.67	41.03
TF-IDF	–	LR	45.66	49.33	47.13
		DT	39.92	37.33	38.15
		RF	42.86	44.00	40.18
		MNB	43.05	40.00	36.06
		SVM	43.84	40.00	34.49
Word2Vec	Base	CNN (C)	59.20	63.15	60.17
		BiLSTM (B)	57.97	47.36	51.47
		GRU (G)	59.18	61.84	59.92
	AE	C + B	58.04	55.26	56.29
		C + G	56.67	59.21	57.09
		B + G	53.80	53.94	53.42
		C + B + G	55.24	57.89	55.98
	WE	C + B	56.06	56.57	56.01
		C + G	56.67	59.21	57.09
		B + G	55.31	56.57	55.49
		C + B + G	55.23	57.89	55.98
FastText	Base	CNN (C)	61.06	64.47	62.63
		BiLSTM (B)	65.69	68.42	66.78
		GRU (G)	58.74	61.84	59.93
	AE	C + B	63.05	67.10	64.87
		C + G	61.30	64.47	62.27
		B + G	64.19	68.42	66.01
		C + B + G	63.83	68.42	65.77
	WE	C + B	65.83	69.73	**67.57**
		C + G	61.32	64.49	62.25
		B + G	63.03	67.12	64.85
		C + B + G	63.80	68.44	65.72

AE, WE indicated average and weighted ensemble respectively

Table 6 Performance comparison between the proposed and existing techniques for textual emotion classification

Methods	EBEmoD f_1-score	BYCD f_1-score
Tripto et al. [29]	53.54%	51.47%
Rahman et al. [22]	57.85%	34.49%
Pal et al. [20]	57.49%	47.13%
Proposed (C + B)	**62.46**	**67.57**

the dependencies in the word sequences. Furthermore, the WE technique readdresses the softmax probabilities based on the primary outcomes of the models. The consideration of previous outcomes surely helps the model to classify text instances more accurately.

Comparison with Existing Techniques

The performance of existing techniques [20, 22, 23, 29] are investigated on the developed dataset (i.e., EBEmoD) and BYCD. For consistency, these techniques are implemented on both datasets and compared their performance with the proposed technique (i.e., C + B). Table 6 shows the results of the comparison concerning weighted f_1-score. The comparative analysis exhibits that the proposed weighted ensemble method outperformed the existing techniques by obtaining the highest f_1-score of 62.46% (EBEmoD) and 67.57% (BYCD).

Error Analysis

The results demonstrated that the proposed weighted ensemble performed better compared to other models. A detailed

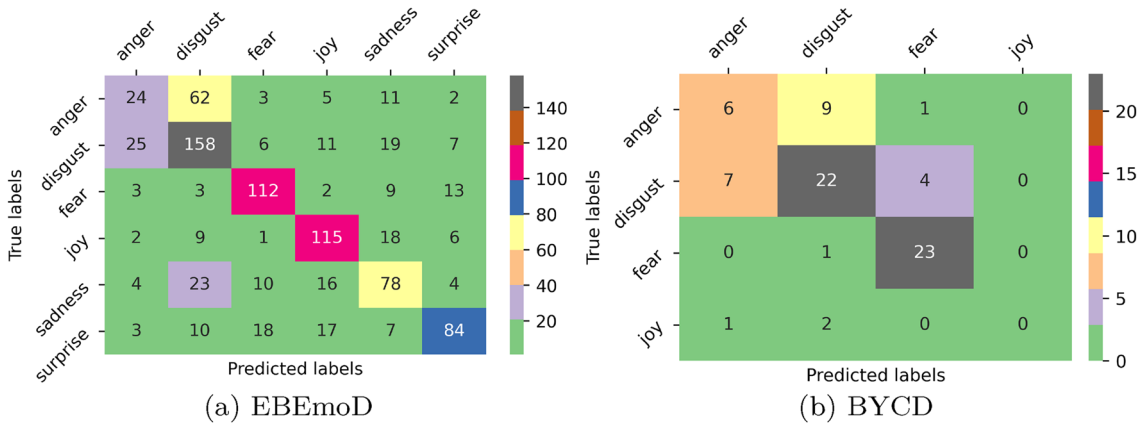

Fig. 3 Confusion matrix of the proposed weighted ensemble model

Fig. 4 Examples indicating contrasting nature of the models. Here, ensemble indicates the proposed model (C + B)

Text	CNN	GRU	BiLSTM	Ensemble	Actual
পাকিস্তানের থেকে স্বাধীন হয়ে ভারতের কাছে পরাধিন আমরা (We are independent of Pakistan and dependent on India)	anger	sadness	disgust	anger	anger
যদি লক ডাউন করা হয় এই মানুষগুলো না খেয়ে মরবে (If locked down is applied, these people will die without eating)	fear	surprise	sadness	sadness	sadness
মুভিটা মুক্তির আগে যতটা এক্সাইটেড ছিলাম দেখার পরে ঠিক ততটাই ডিসাপয়েন্টেড (Just as excited as I was before watching the release of the movie, so dispensed with after watching it)	joy	joy	joy	joy	disgust
লোকটি হঠাৎ করে মারা গেলেন, তার মৃত্যু টা সাধারণ মৃত্যু না (The man died suddenly, his death was not an ordinary death)	sadness	fear	fear	sadness	surprise

error analysis is carried out to get a close look at the proposed model's performances. Figure 3a shows the confusion matrix for EBEmoD. It is observed that disgust, joy, and fear classes correctly classified 158, 115, and 112 among 226, 151, and 142 instances. Out of 135 sad texts, 57 are misclassified. In the case of the anger class, the model performed poorly. It incorrectly classified 83 texts among 107 texts. Figure 3b indicates that disgust and fear classes correctly classified 22 and 23 instances for BYCD. On the contrary, the proposed model incorrectly classified ten instances from 16 samples of anger class. With the shortage of text samples of joy class in BYCD, the predicated model cannot identify any instances correctly.

Furthermore, the performance of the proposed ensemble technique is qualitatively analyzed. The outputs of the individual models and the ensemble model are examined closely. Figure 4 shows few examples illustrating the contrasting nature of the models. We observe mixed predictions from the classifiers in the first two examples where the proposed model correctly classifies. However, in the latter two instances, the model confuses disgust with joy and incorrectly classifies surprise as sadness. The presence of mixed and neutral emotion words might cause this error. In addition, few words are frequent on multiple classes such as joy & surprise, anger & sadness classes have considerable overlap concerning frequent words. The usage of such words confuses the models hence resulted in poor performance. Contextual analysis of the texts with more training data might help the models to tackle such issues.

Conclusion

This paper investigated the performance of various machine learning (LR, DT, RF, MNB, SVM) and deep learning (CNN, BiLSTM, GRU) techniques for the textual emotion categorization task in Bengali. After experimental evaluations of all models, this work proposed a weighted ensemble-based technique (combination of CNN and BiLSTM) to categorize textual emotions in Bengali for its superior performance. This work offers a manually annotated Bengali emotion corpus containing 9000 texts concerning six categories (i.e., anger, disgust, fear, joy, sadness, surprise). Performance analysis exhibits that the proposed weighted ensemble of CNN and BiLSTM (i.e., C + B) provides superior results among all the models. The proposed technique also outperformed the average ensemble and other baselines by obtaining a maximum

weighted f_1-score of 62.44% and 67.57% on EBEmoD and BYCD. In future, it will be interesting to investigate how the proposed model responds if multi-domain heterogeneous data are considered. Moreover, texts from other categories such as hate, stress, love, and texts expressing mixed-emotion can also increase model generalization capability.

Acknowledgements This work is conducted under the ICT Fellowship Program of ICT Division, Ministry of Posts, Telecommunications and Information Technology, Bangladesh.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abdullah M, Hadzikadicy M, Shaikhz S. Sedat: sentiment and emotion detection in Arabic text using cnn-lstm deep learning. In: 2018 17th IEEE international conference on machine learning and applications (ICMLA), pp. 835–840. IEEE (2018)
2. Alswaidan N, Menai MEB. A survey of state-of-the-art approaches for emotion recognition in text. Knowledge Inf Syst 62(8) (2020).
3. Alzu'bi S, Badarneh O, Hawashin B, Al-Ayyoub M, Alhindawi N, Jararweh Y. Multi-label emotion classification for arabic tweets. In: 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 499–504. IEEE (2019).
4. Baali M, Ghneim N. Emotion analysis of Arabic tweets using deep learning approach. J Big Data. 2019;6(1):1–12.
5. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. Trans Assoc Comput Linguist. 2017;5:135–46.
6. Cohen J. A coefficient of agreement for nominal scales. Educ Psychol Measur. 1960;20(1):37–46.
7. Das A, Iqbal MA, Sharif O, Hoque MM. Bemod: development of Bengali emotion dataset for classifying expressions of emotion in texts. In: Vasant P, Zelinka I, Weber GW, editors. Intelligent computing and optimization. Cham: Springer International Publishing; 2021. p. 1124–36.
8. Das A, Sharif O, Hoque MM, Sarker IH. Emotion classification in a resource constrained language using transformer-based approach. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, pp. 150–158. Association for Computational Linguistics, Online (2021). https://doi.org/10.18653/v1/2021.naacl-srw.19. https://aclanthology.org/2021.naacl-srw.19
9. Das SD, Basak A, Dutta S. A heuristic-driven uncertainty based ensemble framework for fake news detection in tweets and news articles. CoRR abs/2104.01791 (2021). arxiv:2104.01791.
10. Ekman P. An argument for basic emotions. Cogn Emotion. 1992;6(3–4):169–200.
11. Garcia-Garcia JM, Penichet VM, Lozano MD. Emotion detection: a technology review. In: Proceedings of the XVIII international conference on human computer interaction, pp. 1–8 (2017).
12. Haryadi D, Kusuma GP. Emotion detection in text using nested long short-term memory. Int J Adv Comput Sci Appl. 2019;10(6):11480.
13. Hossain E, Sharif O, Hoque MM, Sarker IH. Sentilstm: a deep learning approach for sentiment analysis of restaurant reviews (2020).
14. Kim Y. Convolutional neural networks for sentence classification. emnlp (2014).
15. Lai Y, Zhang L, Han D, Zhou R, Wang G. Fine-grained emotion classification of Chinese microblogs based on graph convolution networks. World Wide Web. 2020;23(5):2771–87.
16. Lora SK, Jahan N, Antora SA, Sakib N. Detecting emotion of users' analyzing social media bengali comments using deep learning techniques. In: 2020 2nd International Conference on Advanced Information and Communication Technology (ICA-ICT), pp. 88–93. IEEE (2020).
17. Lx Luo. Network text sentiment analysis method combining lda text representation and gru-cnn. Pers Ubiquitous Comput. 2019;23(3):405–12.
18. Mamta Ekbal A, Bhattacharyya P, Srivastava S, Kumar A, Saha T. Multi-domain tweet corpora for sentiment analysis: resource creation and evaluation. In: LREC (2020).
19. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119 (2013).
20. Pal A, Karn B. Anubhuti—an annotated dataset for emotional analysis of Bengali short stories. arXiv preprint arXiv:2010.03065 (2020).
21. Rahman M, Haque S, Saurav ZR. Identifying and categorizing opinions expressed in Bangla sentences using deep learning technique. Int J Comput Appl. 2020;975:8887.
22. Rahman M, Seddiqui M, et al. Comparison of classical machine learning approaches on Bangla textual emotion analysis. arXiv preprint arXiv:1907.07826 (2019).
23. Rayhan MM, Al Musabe T, Islam MA. Multilabel emotion detection from bangla text using bigru and cnn-bilstm. In: 2020 23rd International Conference on Computer and Information Technology (ICCIT), pp. 1–6. IEEE (2020).
24. Seyeditabari A, Tabari N, Zadrozny W. Emotion detection in text: a review. arXiv preprint arXiv:1806.00674 (2018).
25. Sharif O, Hoque MM, Hossain E. Sentiment analysis of Bengali texts on online restaurant reviews using multinomial naïve bayes. In: 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), pp. 1–6 (2019). https://doi.org/10.1109/ICASERT.2019.8934655.
26. Sharif O, Hossain E, Hoque MM. Techtexc: Classification of technical texts using convolution and bidirectional long short term memory network. CoRR abs/2012.11420 (2020). arxiv:2012.11420.
27. Soleymani M, Garcia D, Jou B, Schuller B, Chang SF, Pantic M. A survey of multimodal sentiment analysis. Image Vis Comput. 2017;65:3–14.
28. Tokunaga T, Makoto I. Text categorization based on weighted inverse document frequency. In: Special Interest Groups and Information Process Society of Japan (SIG-IPSJ. Citeseer (1994).
29. Tripto NI, Ali ME. Detecting multilabel sentiment and emotions from bangla youtube comments. In: 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pp. 1–6. IEEE (2018).
30. Zhang Y, Jin R, Zhou ZH. Understanding bag-of-words model: a statistical framework. Int J Mach Learn Cybern. 2010;1(1–4):43–52.