Multimedia Tools and Applications (2024) 83:29399–29417
https://doi.org/10.1007/s11042-023-16833-5

Sentiment classification with modified RoBERTa and recurrent neural networks

Ramalingaswamy Cheruku[1] · Khaja Hussain[1] · Ilaiah Kavati[1] · A. Mallikarjuna Reddy[2] · K. Sudheer Reddy[2]

Received: 10 October 2022 / Revised: 30 May 2023 / Accepted: 31 August 2023 /
Published online: 12 September 2023

Abstract
The unprecedented growth in the use of social media platforms, where opinions and decisions are made and updated within seconds. Hence, Twitter is becoming a huge commercial interest for brands and companies to assess the sentiment of customers. Sentiment analysis tries to extract subjective opinions and sentiments from opinionated data using Natural Language Processing (NLP). Ontology-based analysis was primarily used to disambiguate the terms and get a high precision score for the emotive terms. In this paper, to improve the accuracy of sentiment analysis we modified the RoBERTa model to extract the more relevant contextualized information. Moreover, this modified RoBERTa is combined with RNN for effective sentiment classification. The proposed work attempts to find which words or phrases actually contribute to the particular sentiment as output by a modified RoBERTa model and this output of the RoBERTa model is fed as input for RNNs. The proposed model is experimented on the Twitter comment dataset. The proposed model experimented on various models such as single RNN, single layer LSTM, and Bi-directional LSTM and evaluated performance measures in terms of accuracy, precision, recall, and F1-score. Our proposed model performance significantly improved with respect to all other models in terms of accuracy, precision, recall, and F1-score. The experiments show that the proposed model not only increases the Jaccard similarity score but also improves different RNN performance when compared to existing state-of-the-art models. The proposed approach obtained a maximum accuracy of 84.6% which is a huge improvement and also evaluated comparison analysis of Simple RNN, Single-LSTM, and Bi-LSTM on full text and selected test. Our proposed modified ROBERTa performance is superior with selected text and full text. Finally, the statistical paired T-test is performed between the proposed model, and other models such as simple RNN, One layer RNN is giving evidence that the proposed model performance is superior with 95% confidence and $p < 0.05$.

Keywords Sentiment analysis · Context understanding · Natural language processing · RoBERTa · Recurrent neural networks

✉ Ramalingaswamy Cheruku
rmlswamy@nitw.ac.in

Extended author information available on the last page of the article

1 Introduction

The extraction and analysis of new patterns that emerge as a result of the quick growth of social media groups is made possible by sentiment analysis and opinion mining [1]. Sentiment analysis typically uncovers the sentiment orientation (positive, neutral, or negative) of textual data. This information can help decision-makers in a variety of industries, including finance and the stock market [2–4], digital payment services [5].

Emotion analysis aids in our comprehension of and retrieval of any potential emotional data included in any user-generated content. Text, photos, animations, movies, scan images, and other types of material can be created by users. The endeavour of extracting emotions from a text document is difficult since there are many ambiguities and inconsistencies that permeate the text's substance. Different approaches, including the corpus-based model, the appraisal-based model, and the knowledge-based model, have been used to classify emotion analysis. As an illustration, the word "eager" has a higher probability score than "curious" or "willing." An ontology-based analysis has been primarily employed to disambiguate the terms and to produce a high accuracy score for the emotional terms in order to avoid these challenges [6].

Ontology was once restricted to psychology, but starting in the 1980s, it began to gain traction in computer science. It has become increasingly popular in recent years to conduct research ontology-based emotion extraction on text materials. Ontologies have been employing formal language representations to describe the domains of the input document, just as humans always use natural language to define the domain of the specific text content. Numerous academic fields, including artificial intelligence, entity extraction, Semantic Web, collaborative software development, and many more, have made extensive use of ontologies. Huge advantages of ontologies include conceptualization, reuse, resource sharing, and term coreferencing [6].

As previously stated, it is a cost-effective approach for dealing with text analysis and is quite robust for many application areas. Many research projects have used ontologies to remove the ambiguity that exists in textual content. A structure that provides a formal definition of a standard representation of real-world concepts can aid in their comprehension. COVID-19 is a coronavirus disease caused by SARS-CoV-2 that surfaced in December 2019 (6-11). Due to a lack of previous research, we are able to investigate this ontology-based emotion extraction on COVID-19 datasets [6].

Both the scientific community and the corporate sector have expressed interest in the possibility of automatically capturing the general public's opinions about social events, political movements, marketing campaigns, and product preferences [7]. Sentiment analysis using machine and deep learning has numerous obstacles, including a lack of labeled data and weak generalization capacity. To increase model performance, researchers coupled sentiment knowledge with supervised data. Sentiment knowledge based on quality background dictionaries, for example, can capture fine-grained supervision information when extracting and evaluating sentiments from texts [8, 9]. Because sentiment dictionary knowledge is included into the modeling language, a word vector representation increases sentiment analysis task performance [10].

Traditional machine learning methods such as Naive Bayes, Decision Trees, and Support Vector Machines are used in sentiment analysis algorithms, as are deep learning models such as Recurrent Neural Networks (RNNs) [11]. When analyzing long sequences of data, such as text, RNNs suffer from vanishing gradients difficulties. When the gradients used to update the parameters of the RNN during training become extremely small as they travel

backward through time, the vanishing gradient problem develops. This might lead to delayed or ineffective learning since parameter updates become insignificant and the RNN fails to grasp long-term dependencies in the input. In deep networks, RNN networks are prone to causing gradient explosion or vanishing gradient problems. However, the GRU and LSTM structures have been frequently used in NLP research. Bidirectional network architectures, in particular, have been shown in research to be more effective at learning text context information. As a result, bi-LSTM [12] and bi-GRU [13] have been frequently used in recent studies [14].

Petrucci [14] proposed recently, Neural Networks have been proven extremely effective in many natural languages processing tasks such as sentiment analysis, question answering, or machine translation. Aiming to exploit such advantages in the Ontology Learning process, in this technical report we present a detailed description of a Recurrent Neural Network based system to be used to pursue such a goal. The biLSTM/biGRU structure alone is computationally intensive. Through the gating mechanism, the LSTM can slow down the problem of vanishing gradient and gradient explosion in RNN. Because the level is too deep, backpropagation will encounter vanishing gradient and explosion issues. In the suggested model, the complex deep BiLSTM structure is replaced with the Enhanced Multi-Head Self-Attention shallow network to prevent potential gradient concerns. Our contributions are

1. modified RoBERTa model for extracting more contextualized information from Twitter sentiments.
2. outcome of modified RoBERTa model is fed into RNNs for effective classification of Twitter sentiments.

Social media platforms have become important sources of information and opinions. This massive volume of information is critical for understanding the interactions and dynamics of subjectivity on the Internet, which is particularly important for marketing purposes. Twitter is one of the most popular micro-blogging platforms, with over a billion active users and 500 million daily messages [15]. However, analyzing such a large amount of data remains difficult due to the informal nature of language, which is influenced by typos and defined by slang. It gave rise to what is currently known as Sentiment Analysis, a set of tasks aiming to detect the subjective attitude of a writer with respect to some topic. The fundamental goal of sentiment analysis is to categorize opinions into different perspectives. The polarity of sentiment, on the other hand, is greatly reliant on the context of the sentiment text. In order to appropriately classify sentiments, it is necessary to consider contextual information [16].

The overall organization of the paper is as follows Section 2 discussed the motivation of the work and Section 3 relates the work Section 4 is preliminaries Section 5 proposed models Section 6 is results and discussion followed by a conclusion.

2 Motivation

The majority of previous research focuses on feature engineering based on the content of tweets. They primarily concentrate on statistical aspects such as Bag-of-words and TF-IDF(Term Frequency-Inverse Domain Frequency), which ignore sequence and context. Pre-trained language models, such as BERT [17] and RoBERTa [18], have proven to be useful for a variety of applications in recent years. Text sentiment may be detected and classified using such pre-trained models. With the help of an autonomously created ontology comprising domain-specific common-sense information, we proposed a technique that picks

only important features and elements about which any opinion is expressed. Furthermore, this ontology is used to extract only domain-specific concepts.

3 Related work

Various literature has reported different approaches in sentiment analysis, ranging from lexicon-based to machine-learning approaches. However, not many kinds of literature deliberate on a context-based approach for sentiment analysis. Currently, researchers widely use many canonical NLP features, such as TF-IDF, topics, syntactic, affective characteristics, readability, and deep learning models, such as CNN and LSTM. Those methods, especially DNNs with automatic feature learning, boosted predictive performance and preliminary success on suicidal intention understanding.

Tan et al. [19] proposed a model it combines a robustly optimized BERT method and Long Short-Term Memory. The Robustly Optimised BERT technique compactly maps words into a meaningful word embedding space, but the Long Short-Term Memory model efficiently captures long-distance contextual semantics. The experimental results show that the proposed hybrid model outperforms state-of-the-art approaches, with F1 scores of 93%, 91%, and 90%, respectively, on the IMDb dataset, Twitter US Airline Sentiment dataset, and Sentiment140 dataset.

Monika et al. [20] proposed deep learning approaches to investigate word embedding models (Word2Vec, Glove) in tweets to detect sentiment polarity. We examined sentiment analysis utilizing the Recurrent Neural Network (RNN) model in conjunction with Long-Short Term Memory (LSTM) units, which can deal with long-term relationships by introducing memory in a network model for prediction and visualization. The results demonstrated that our models are dependable for future prediction, with considerable classification accuracy trained at 80% for the training set and 20% for the testing set. For future study studies, the Bidirectional LSTM Model (Bi-LSTM) is applied to increase this performance.

Sivasai et al. [21] proposed study, performed pre-processing utilizing the adaptive bilateral filter (ABF) to remove noise from an MR image. The binary thresholding and Fuzzy Recurrent Neural Network (FR-Net) segmentation algorithms were then used to reliably detect the tumor region. Datasets for training, testing, and validation are used. We will use our machine to predict whether or not the subject has a brain tumor. The outcomes were evaluated using several performance indicators such as accuracy, sensitivity, and specificity.

Nair et al. [16] has implemented 3 different algorithms for Covid-19 tweets. They have implemented Logistic regression, BERT, and VADER for sentiment analysis, keeping the preprocessing steps the same. The proposed methods are more sensitive to sentiment expressions. BERT is reported to have high accuracy among the 3 algorithms because of capturing context in both forward and backward directions.

Bhuvan et al. [22] used Naive Bayes, Support Vector Machines, and Logistic Regression models for the classification of the movie review dataset. They have proposed a new grammar model where regular expressions for each positive and negative review. This model is context-specific and Logistic Regression gave the highest accuracy with SGD on Apache Spark.

Leo et al. [23] created a new architecture by fusing hidden layers of BERT with word embeddings such as ELMo using GRUs. They have observed the linguistic information present across hidden layers of BERT and tried to exploit this fact. Their model can be applied to other BERT-based models such as Roberta. The model has been prevented from overfitting by using early stopping and voting classifier.

Katz et al. [24] proposed a new approach ConSent.To find important phrases indicative of the presence of sentiment, they have used approaches from the field of information retrieval. The context words are used to represent the relationships between the key terms that have been found in order to determine the best appropriate context on every central theme. Their model has strength against noise and has performed well compared to state-of-the-art models.

Tang at al. [25], argued that word embeddings and context-based word embeddings are not the best if applied for sentiment classification. They built a sentiment lexicon and mapped neighboring words that encode emotional texts in continuous word representation to sentence and word-level sentiment analysis. The proposed lexicon is based on sentiment embedding that is useful for improving lexical-level tasks in finding similarities between words.

Vimal and Murugan [26] used skip-grams and Continuous Bag of Words(CBOW) for feature representation. They took each word embedding and passed it to BiLSTM to get the text illustration feature. Later F-LSTM and Backward B-LSTM are merged. Finally, Softmax is applied. The Amazon reviews dataset has given 90.46% accuracy on this model.

Vaswani et al. [27] introduced a new neural network architecture called transformer based on the self-attention mechanism. Attention is a concept that helped improve the performance of neural machine translation applications. Transformer is a model that uses attention to boost the speed with which these models can be trained. This architecture has performed better than recurrent and convolutional models in language understanding tasks such as English-German translation.

Based on the Transformer encoder architecture, Devlin et al. [17] proposed BERT. BERT takes into consideration both the preceding and subsequent data, employing the Self-attention Mechanism to account for the interdependence of words in sentences. The BERT was made up of stacked transformer encoders, and the paper proposes varying layer counts for various configurations Tables 1 and 2.

4 Preliminaries

4.1 BERT

Recent advancements in NLP has shown that pre-trained Language Models (LM) has given better results in various tasks [28]. Language modeling is a probabilistic density estimation problem. It is a very popular challenge in NLP.

Table 1 Limitations of previous studies (gaps)

S. No	Limitation	Reference
1	Ensemble-based hybrid model with Computational overhead	[19]
2	Performance can be improved with Bi-LSTM	[20]
3	domain-based and authors have not looked towards the mood of the user work	[16]
5	Defining grammars is an overhead of the proposed model. specific to the context	[22]
6	There should be considerable computational overhead	[23]
7	adapt proposed method to multi-class problems	[24]
8	proposed work cannot handle the words not covered in the embedding vocabulary	[25]

Table 2 Confusion matrix of Bi-LSTM with three sentiment labels of Neutral, Negative, and Positive

NEUTRAL	2471	181	81
NEGATIVE	447	1487	32
POSITIVE	323	233	1615
	NEUTRAL	**NEGATIVE**	**POSITIVE**

Given a sequence of words, $w_{1:n} = [w_1, w_2, w_3 \ldots, w_{n-1}, w_n]$, its joint probability $P(w_{1:n})$ is expressed as:

$$P(w_{1:n}) = \prod_{t=1}^{n} P(w_t \mid W_{0:t-1}) \tag{1}$$

where W_0 denotes the start of the input sequence.

One disadvantage of traditional single-directional language models is that they capture context only in leftward tokens and themselves. But capturing context in both directions is important. To overcome this, Devlin et al. proposed BERT, a new pre-training model in the year 2018 [17]. BERT masks certain tokens from the input phrases before training the model to predict the masked tokens based on the remaining tokens. To do masking, Devlin utilized a different token [MASK] 80% of the time, a random token ten percent of the time, and unchanged for another ten percent of the time.

4.2 RoBERTa architecture

RoBERTa relies on the language masking method of BERT model, which trains the system to predict purposely hidden content within otherwise unannotated language instances. RoBERTa modified some important hyperparameters in the BERT model. BERT's next-sentence pretraining target has been deleted. RoBERTa is trained with considerably bigger-sized mini-batches, adopted dynamic masking [18], and learning rates. After these changes, RoBERTa has shown better results than BERT on the masked language modeling target.

5 Proposed methodology

In this paper, we have

1. modified RoBERTa model for extracting more contextualized information from Twitter sentiments.
2. outcome of modified RoBERTa model is fed into RNNs for effective classification of Twitter sentiments.

The overall architecture of the proposed approach is shown in Fig. 1 and steps as follows

1. pre-processing of raw tweets to generate tokens
2. tokens are fed to modified pre-trained ROBERTa to generate tensor outputs.
3. With tensor outputs module is fine-tuned.
4. finally selected text is passed to RNN for classification.

5.1 Proposed modified RoBERTa model

We have added following custom question answer head layers to pre-trained RoBERTa model:

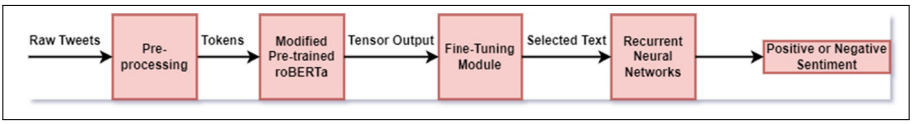

Fig. 1 Overall Framework of Proposed Model

1. Masked self Attention Layer: To bake the "understanding" of other relevant words into the one currently being processed in longer sequences.
2. Dropout Layer: To avoid the model from overfitting
3. Conv1D Layer: To convert the word embeddings into a specific form.
4. Relu Layer: To convert embeddings into non-linear. ReLU is a simple activation function. It is a less expensive and non-linear function that will output the input the same as the original if it is positive, else it outputs 0.

$$\mathcal{R}(z) = max(0, z) \tag{2}$$

Using the max() function over the set of 0.0 and the input z, we may mathematically express this function g(); for illustration: The function has many of the desirable Characteristics of a linear activation function for training a neural network using backpropagation since it is linear for values larger than zero. However, because negative values are always output as zero, it is a nonlinear function.

5. Dense Layer: It is a feed-forward neural network.
6. Flatten Layer: Convert into 1-dimensional data.
7. Soft max Layer: It is for multi-class classification. The purpose of softmax is to construct a probability distribution with the same number of components as a vector of real numbers, with bigger elements having greater probabilities and smaller elements having lower probabilities.

$$\sigma(g_l) = \frac{e^{z_l}}{\sum_{p=1}^{m} e^{g_p}} \quad for\ l = 1, 2, 3, 4 \ldots, m \tag{3}$$

where

(a) σ= Softmax
(b) gl= input vector
(c) e_l^z = standard exponential function for an input vector
(d) m=number of classes in the multi-class classifier
(e) e^{g_p}=standard exponential function for output vector

The modified RoBERTa model is shown in Fig. 2

5.2 Recurrent neural networks

The outcome of the modified RoBERTa model is fed into RNNs. The RNNs or Recurrent neural networks are used for classification in this paper. RNNs have a temporal aspect in RNN. RNNs vary from feed-forward neural networks because of their time aspect.

A recurrent neural network (RNN) is a class of neural networks that allow previous outputs to be used as inputs while having hidden states. Thus RNNs show temporal dynamic behavior. Figure 3 shows traditional RNN. At timestep s, the activation $u^{<s>}$ and the output $o^{<s>}$are represented as:

$$u^{<s>} = A_1(X_{uu}u^{s-1} + X_{ux}x^{<s>} + b1_u \quad \text{and} \tag{4}$$

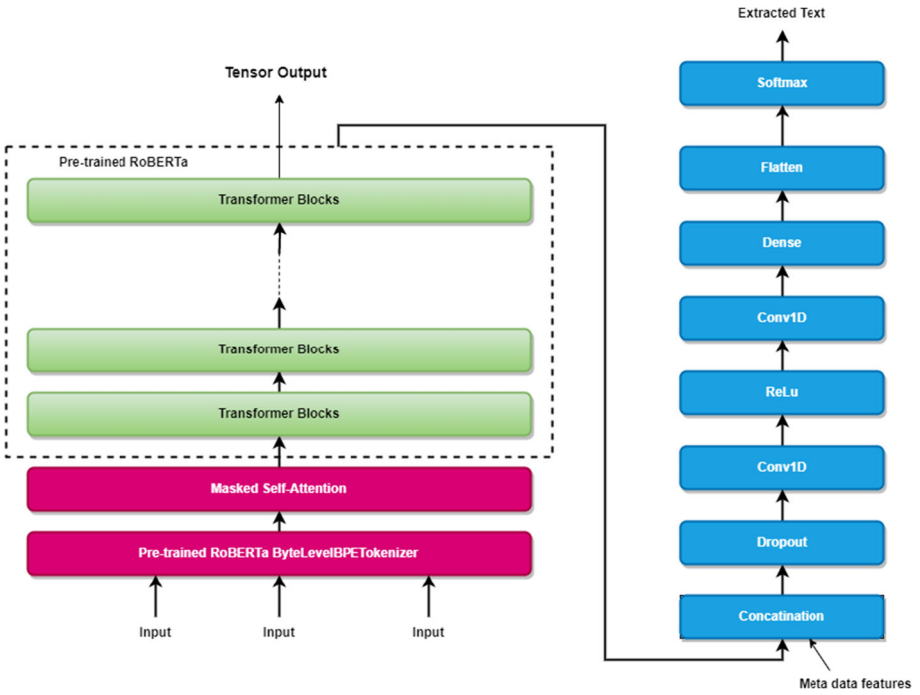

Fig. 2 Proposed model

$$o^{<s>} = A_2(X_{ou}u^s + b2_o) \tag{5}$$

where W_{ux}, W_{uu}, W_{ou}, $b1_u$, $b2_o$ are the coefficients which are temporally shared and A_1, A_2 are activation functions.

Each block inside a single RNN unit is shown in Fig. 4. However, traditional RNN has a vanishing gradient problem, which is dealt with by Long Short-Term Memory units (LSTM). It maintains long-range connections. For the current sentiment classification problem, we have used the many-to-one architecture of RNNs.

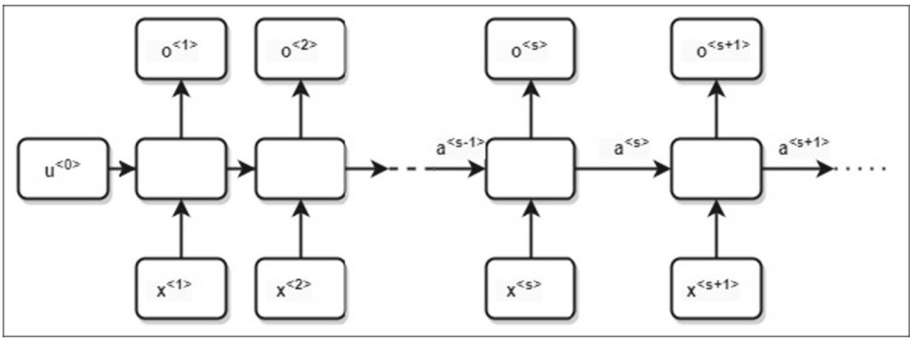

Fig. 3 Traditional RNN

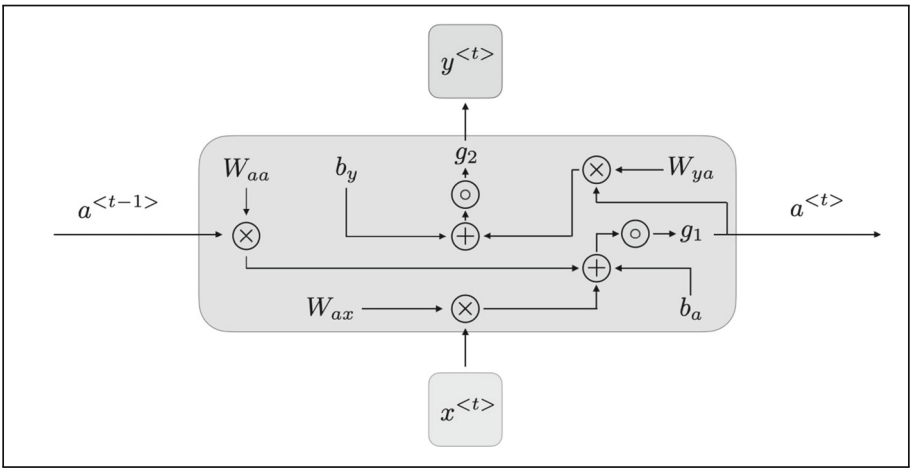

Fig. 4 Inside Simple RNN block

In contrast to LSTM, BiLSTM accepts input in both directions and utilizes data from both sides. In both directions, it accurately simulates sequential relationships between phrases and words. Figure 5 depicts BiLSTM's basic architecture. It creates a new layer with input flowing backward also. We have used the sigmoid activation function in all three RNNs. It is defined as

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{6}$$

A mathematical function with a distinctive "S"-shaped curve, also known as a sigmoid curve, is called a sigmoid function here

1. $S(x)$ = sigmoid function
2. e = Euler's number

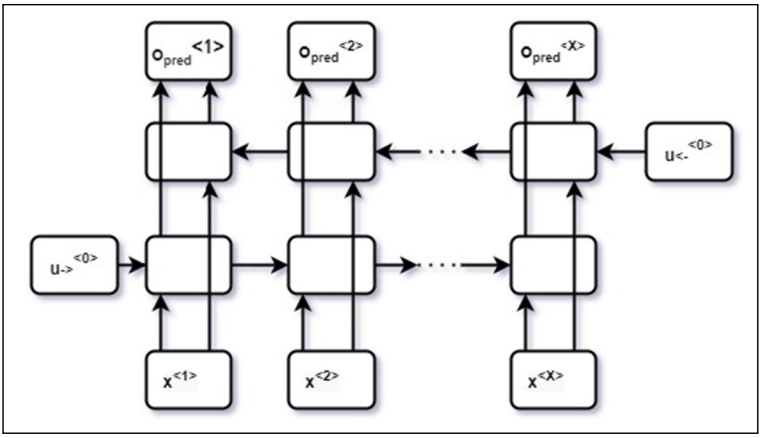

Fig. 5 Bi-directional RNN

Each RNN is fed with original full text and predicted output from RoBERTa i.e., selected text as input separately along with sentiment label to each RNN. The loss in RNNs is calculated at every time step t, using loss function $\mathcal{L}$ is calculated as follows:

$$\mathcal{L}(o_{pred}, o) = \sum_{t=1}^{X_o} \mathcal{L}(o_{pred}^{<t>}, o^{<t>}) \tag{7}$$

Each RNN in our experiments is designed by adding following layer:

- An embedding layer to convert input into dense vectors
- RNN layer
- A regular densely connected neural network layer

Performance metrics are evaluated and presented in Results and Discussion.

6 Results and discussion

6.1 Implementation platform

Google Colab was used to perform experiments. All the experimental results and results analysis is carried out using Python, PyTorch, PyTorch transformers, and the sklearn library.

6.2 Dataset

In our study, we have used the Twitter dataset from the Kaggle repository consists of tweets and their corresponding sentiment labels and selected sentiment-bearing phrases. Here is a description of the dataset components:

textID: A unique identifier for each tweet. text: The text content of the tweet. sentiment: The sentiment label of the tweet, which can be either "positive," "negative," or "neutral." selected_text: The selected text from the tweet that represents the sentiment. This column is only available in the training set. There are 31,015 tweets in our study we have trained with 27481 and tested with 3534 and also maximum words, the maximum length of a sentence is 128. We try to predict a word or phrase from each tweet that represents the sentiment expressed. This word or phrase should encompass all the characters within that span, including any punctuation marks or spaces. The expected format for predictions is as follows:

<id>,"<predicted word or phrase that reflects the sentiment>"

- Text ID = f87dea47db
- comment = what interview! leave me alone
- selcted_text = leave me alone
- label = negative.

6.3 Pre-processing and tokenization

Scraped tweets from Twitter usually provide a noisy dataset. This is due to the informal nature of people's social media usage. Retweets, emotions, user mentions, and other unique aspects of tweets must be retrieved appropriately. As a result, raw Twitter data must be standardized to build a dataset that various classifiers can readily learn. To normalize the dataset and decrease its size, we used pre-processing steps.

To make input ready to be trained on the RoBERTa model, we have used ByteLveleBPETokenizer [29]. BPE divides the training data into words using a pre-tokeniser. Space could be as simple tokenization. After pre-tokenization, a set of unique words was formed and frequency of each word was calculated. BPE then creates a base vocabulary consisting of all symbols discovered in the set of unique words, and tries to merge two symbols to create a new symbol until fixed vocabulary size is obtained. Merge rules are learned in this process. Vocabulary size has been set to 50,265 and symbols learned with 50,000 merges.

We start by pre-processing tweets in general, as follows:

1. Make a copy of original text and selected text and add space at the beginnings
2. Substitute space for two or more dots (.)
3. Add space between each pair of special characters ".", "!" and "?"
4. Use a single space to replace two or more spaces.
5. Encode both text and selected_text. Find sublist occurrence of selected_text in text.
   (a) if sub-list is not empty, entire text is appended to selected_text along with space at beginning.
   (b) if sub-list is empty, the particular tweet is dropped because it is neutral class.

6.3.1 Label encoding

The labels in our dataset are positive, negative, and neutral. We must transform them to a float type because model cannot comprehend categorical nature of the dataset. The to_categorical function from Keras library will be used to accomplish this operation.

6.3.2 Data sequencing and splitting

Our neural networks need tokens input to work with. Tokenizer from Keras library is utilized to convert our text input into 3D float datatype. pad_sequences function from Keras is used for padding our tokens data.

6.4 Evaluation metrics

Confusion Matrix: Confusion matrix is a tabular way of visualizing the performance of a prediction model. Each entry in a confusion matrix denotes the number of predictions made by the model where it classified the classes correctly or incorrectly of the classification model as follows:

- True Positive (TP) represents number of sentences rightly classified as specific class.
- False Negative (FN) is the sum of values of corresponding rows except for the TP value
- False Postive (FP) is the sum of values of the corresponding column except for the TP value.
- True Negative (TN) is the sum of values of all columns and rows except the values of that class that we are calculating the values for.

For better understanding purpose, a confusion matrix of Bi-LSTM which is generated on full-text data is shown below.

For instance, negative class,

- TP_negative = 447
- FP_negative = FN_neutral + FN_positive = 2471 + 323 = 2794
- FN_negative = FP_neutral + FP_positive = 1487 + 32 = 1519
- TN_negative = Total - (TP_negative + FP_negative + FN_negative) = 6870 - (447 + 2794 + 1519) = 2110

With TP, TN, FP, and FN values we can compute all the performance measures such as Accuracy, Recall, and F1-score using the below-given formulas.

- **Accuracy:** Its accuracy will be calculated to assess the performance of the classification model as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

- **Precision:** It is the measure of out of all positive values predicted, how many are actually true positives and it is defined as follows,

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

- **Recall:** It is also called as sensitivity and it is a measure of out of all correctly classified , how many are actually true positive. It is formulated as,

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

- **F1-score:** It is an important measure which combines both precision and recall to show the performance for our prediction task. It is simply harmonic mean of both (9) and (10).

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{11}$$

- **Jaccard measure:** It calculates how similar the two sets of text data are. It takes ratio of count of intersection of all words in both data sets to count of union of all words in both data sets. It's value ranges from 0 to 1.

$$\mathcal{J}(S1, S2) = \frac{|\ S1\ \cap\ S2\ |}{|\ S1\ \cup\ S2\ |} \tag{12}$$

where, $\mathcal{J}$=Jaccard Score
S1=Set 1
S2=Set 2
The metric used for the calculating overall is defined as,

$$overall\_jaccard\_score = \frac{1}{n} \sum_{i=1}^{n} \mathcal{J}(gt_i, dt_i) \tag{13}$$

where, n = number of documents
$\mathcal{J}$ = jaccard function defind in (12)
gt_i = the ith ground truth
dt_i = the ith prediction

6.5 Training RoBERTa model

During the training of the modified RoBERTa model, we have fine-tuned the Adam optimizer to minimize the cross-entropy loss function. The loss function is evaluated between the predicted text of the RoBERTa model and the actual text. For training of the RoBERTa model, we have partitioned the dataset according to 5-fold cross-validation. In each fold, we calculated the similarity between the RoBERTa model outcome and the actual text using the Jaccard score. The fined tuned hyperparameters of the modified RoBERTa model is shown in Table 3.

1. Tokens are input into bert model and we use BERT's first output. These are embeddings of all input tokens and have shape (batch_size, MAX_LEN, 768).
2. We apply Conv1D function from Tensorflow Keras library. Only one filter is applied. Convolution window size is fixed as 1. Now the embeddings are converted into the shape (batch_size, MAX_LEN, 1).
3. To transform the final output from x1 has shape the (batch_size, MAX_LEN), flatten and softmax are applied. These are one hot encoding of the start tokens indicies (for selected text). And x2 are the end tokens indicies.

RoBERTa is a variant of the Transformer-based architecture used for natural language processing tasks, specifically designed for pretraining on large amounts of unlabeled text data. The initial weights or parameters in the RoBERTa model are typically initialized randomly before the training begins.

More specifically, the model's parameters, including the weights of the neural network layers, are usually initialized using a process called "Xavier" or "Glorot" initialization. Xavier initialization is a popular technique that aims to set the initial weights to prevent the gradients from exploding or vanishing during training. It takes into account the input and output dimensions of each layer. It initializes the weights using a Gaussian distribution with zero mean and variance calculated based on the layer's input and output dimensions.

During the training process, the RoBERTa model is pre-trained on a large corpus of unlabeled text using a variant of the masked language model (MLM) objective. This involves randomly masking out certain tokens in the input text and training the model to predict the original masked tokens based on the context. The pre-trained model is then fine-tuned on specific downstream tasks by further training on labeled data with task-specific objectives.

It's important to note that the exact details of the initialization process and training procedure can vary depending on the specific implementation and configuration of the RoBERTa

Table 3 Fine tuned hyperparameters for RoBERTa Model

No	Hyperparameter	Value
1	maximum length of predicted sequence	108
2	size of encoder layers and pooler layer	768
3	batch size	32
4	number of epochs	5
5	learning rate	2.5e-5
6	dropout probability in encoder, and pooler	0.1
7	dropout ratio for attention probabilities	0.2
8	number of classes	2
9	number of folds	5

Table 4 RNNs parameter values

Parameters	Model Simple RNN	LSTM	Bi-dir LSTM
Input dimension of embedding layer	5000	5000	5000
Output dimension of Embedding layer	15	15	40
Number of RNN units	15	15	20
RNN dropout	0	0.5	0.6
Number of dense layer units	3	3	3
Dense layer activation function	softmax	softmax	softmax

model. Different versions or variations of the model may employ slightly different strategies for weight initialization and training, but the general principles mentioned here provide a common understanding of how initial weights are typically calculated in the RoBERTa model.

Next, the outcome of the modified RoBERTa model is fed into different RNNs namely simple RNN, LSTM, and Bi-directional LSTM. We have applied rmsprop optimizer and categorical cross entropy as loss functions in our RNNs. The hyperparameters for each RNN are set as in the Table 4.

6.6 Results

The Twitter data set is experimented on proposed approach along with other state-of-the-art models for extracting selected text. These models are compared in terms of Jaccard score. In Table 5 shows the performance results of these models. From the table it is observed that proposed model got highest similarity score. It means proposed model extracted more similar text.

Next, each RNN model has been trained against the full text as well as the selected text obtained from the proposed RoBERTa approach. The performance of each model is evaluated on the test data set. The reported accuracies of each model are shown in Table 6. From the table, it is observed that every RNN model has shown better performance while experimenting with extracted text from the proposed approach than full text. The highest performance has been shown by Bi-directional LSTM. The accuracy has been improved in BiLSTM from 68.80% on full text to 84.6% on a selected text by the proposed RoBERTa approach. Also, different evaluation metrics of best-performed model i.e., BiLSTM are reported in Table 7.

Further, we analyzed the performance of each RNN model used in this study in terms of accuracy, precision, recall, and F1-score against full text and selected text (extracted text)

Table 5 Jaccard Score Comparison of proposed model with existing models

Model	Jaccard Score	Year [Ref.]
Transformer	0.608	2021 [30]
BERT-based Model	0.677	2021 [30]
Basic RoBERTa Model	0.708	2021 [30]
Proposed RoBERTa-based Model	**0.733**	

Table 6 Showing performance of RNN models on full text and extracted selected text

Model used for classification	Accuracy on full text	Accuracy on selected text extracted from RoBERTa model
Simple RNN	65.07	81.63
Single Layer LSTM	67.47	84.01
Bi-directional LSTM	68.80	84.60
GRU	70.81	82.09
Bi-GRU	72.50	83.68

respectively. These results are shown in Table 8. From the table, it is clear that using the proposed approach performance of each RNN is improved drastically.

6.7 Discussion

Modified RoBERTa (Modified Robustly Optimized BERT Approach) is a language model based on the BERT (Bidirectional Encoder Representations from Transformers) architecture.

It shares the same transformer-based architecture as BERT but incorporates several improvements that make it superior to Bi-LSTM (Bidirectional Long Short-Term Memory) and Bi-GRU (Bidirectional Gated Recurrent Unit) models. Here are a few reasons why the proposed model is considered as superior:

1. **Pretraining on a massive amount of data:** RoBERTA is pre-trained on a significantly larger corpus of text data compared to the data used for training Bi-LSTM or Bi-GRU models. This extensive pretraining helps RoBERTA learn more generalizable and nuanced language representations.
2. **Bidirectional context:** RoBERTA, like Bi-LSTM and Bi-GRU models, captures contextual information from both directions. However, RoBERTA achieves this using a transformer-based architecture, which allows it to attend to the entire input sequence simultaneously, capturing global dependencies more effectively.
3. **Transformer architecture:** RoBERTA utilizes a transformer-based architecture, which has been shown to be highly effective in modeling sequential and contextual information in the text. Transformers enable parallel processing and attention mechanisms, allowing for efficient and comprehensive learning of dependencies across the input sequence.
4. **Masked Language Modeling (MLM):** BERT and RoBERTA models employ MLM during pretraining. In MLM, a portion of the input text is randomly masked, and the model learns to predict the masked tokens based on the surrounding context. This approach helps RoBERTA learn a deeper understanding of context and improves its ability to fill in the missing information.

Table 7 Performance metrics of best model: Bi-LSTM

Class	accuracy	precision	recall	F1-score
Neutral	70.76%	0.74	0.55	0.63
Negative	80.4%	0.6	0.76	0.67
positive	84.39%	0.69	0.81	0.75

Table 8 Performance improvement of various RNNs using proposed model

Models	Evaluation Metric			
Model performance using full text				
	Accuracy	Precision	Recall	F1-score
Single RNN	0.6991	0.68	0.67	0.67
Single Layer LSTM	0.7358	0.75	0.73	0.74
Bi-directional LSTM	0.7352	0.83	0.74	0.74
Model performance using proposed approach				
Single RNN	0.8110	0.83	0.80	0.81
Single Layer LSTM	0.8458	0.85	0.84	0.85
Bi-directional LSTM	0.8473	0.85	0.84	0.85

5. **Training procedure:** RoBERTA employs a different training approach compared to BERT. It removes the next sentence prediction (NSP) task used in BERT and trains the model on longer sequences. This modification enables RoBERTA to learn more effectively from the available data and improve its performance.

Overall, the enhancements in training data, architecture, and training procedure make RoBERTA a more powerful language model compared to Bi-LSTM and Bi-GRU models. It achieves state-of-the-art performance on various natural language processing (NLP) tasks, such as text classification, question answering, and text generation.

7 Statistical analysis

Before performing the t-test, a number of assumptions must be confirmed, just like with most statistical tests. Understanding the t-test and its variants can help researchers properly plan trials and analyze data with better statistical rigor. We have performed statical analysis using paired T-test [31]. In the statistical analysis, 5-fold cross-validation (5FCV) accuracies are considered for paired T-test. When paired T-test is performed between our proposed model and other models such as Single RNN, and Single Layer LSTM obtained 0.002, and 0.004 probabilities respectively. As $p < 0.05$ our proposed model is differes statistically.

8 Conclusion

Our proposed approach used the Twitter dataset from the Kaggle repository and extracted sentiment phrases. In our model, the modified RoBERTa model for extracting more contextualized information from Twitter sentiments outcome of the modified RoBERTa model is fed into RNNs for effective classification of Twitter sentiments, and pre-processing of raw tweets to generate tokens are fed to modified pre-trained ROBERTa to generate tensor outputs. With tensor outputs module is fine-tuned. Finally, the selected text is passed to RNN for classification. We have performed task-specific fine-tuning of modified RoBERTa model has resulted in better Jaccard scores of 0.733 i.e., the more relevant contextualized text is predicted. The deep learning RNN models while experimenting on selected text obtained from the proposed approach have given much high performance than full text for sentiment anal-

ysis. Thus, the proposed combined strategy of phrase extraction from RoBERTa and RNN has outperformed well than traditional RNN models. Among RNNs, BiLSTM has shown the best performance with an accuracy of 84.6% which is a huge improvement and also evaluated comparison analysis of Simple RNN, Single-LSTM, and Bi-LSTM on full text and selected test. Our proposed modified ROBERTa performance is superior with selected text and full text. Finally, the statistical paired T-test is performed between the proposed model, and other models such as simple RNN, One layer RNN is giving evidence that the proposed model performance is superior with 95% confidence and probability p is less than 0.005.

References

1. Liu B, Zhang L (2012) A survey of opinion mining and sentiment analysis. In: Mining text data, pp. 415–463. Springer, ???
2. de Oliveira Carosia AE, Coelho GP, da Silva AEA (2021) Investment strategies applied to the brazilian stock market: a methodology based on sentiment analysis with deep learning. Expert Syst Appl 184:115470
3. Jing N, Wu Z, Wang H (2021) A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. Expert Syst Appl 178:115019
4. Zhang J, Zhang A, Liu D, Bian Y (2021) Customer preferences extraction for air purifiers based on fine-grained sentiment analysis of online reviews. Knowl Based Syst 228:107259
5. Balakrishnan V, Lok PY, Abdul Rahim H (2021) A semi-supervised approach in detecting sentiment and emotion based on digital payment reviews. J Supercomput 77:3795–3810
6. Narayanasamy SK, Srinivasan K, Mian Qaisar S, Chang C-Y (2021) Ontology-enabled emotional sentiment analysis on covid-19 pandemic-related twitter streams. Front Public Health 1902
7. Cambria E, Das D, Bandyopadhyay S, Feraco A (2017) Affective computing and sentiment analysis. A practical guide to sentiment analysis 1–10
8. Teng Z, Vo DT, Zhang Y (2016) Context-sensitive lexicon features for neural sentiment analysis. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp. 1629–1638
9. Qian Q, Huang M, Lei J, Zhu X (2016) Linguistically regularized lstms for sentiment classification. arXiv preprint arXiv:1611.03949
10. Tang D, Wei F, Yang N, Zhou M, Liu T, Qin B et al (2014) Learning sentiment-specific word embedding for twitter sentiment classification. In: ACL (1), pp. 1555–1565
11. Birjali M, Kasri M, Beni-Hssane A (2021) A comprehensive survey on sentiment analysis: Approaches, challenges and trends. Knowl Based Syst 226:107134
12. Ma J, Ganchev K, Weiss D (2018) State-of-the-art chinese word segmentation with bi-lstms. arXiv preprint arXiv:1808.06511
13. Lerner I, Paris N, Tannier X (2020) Terminologies augmented recurrent neural network model for clinical named entity recognition. J Biomed Inform 102:103356
14. Petrucci G, Ghidini C, Rospocher M (2016) Using recurrent neural network for learning expressive ontologies. arXiv preprint arXiv:1607.04110
15. Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and opinion mining in: Proceedings of the seventh conference on international language resources and evaluation. European languages resources association, Valletta, Malta
16. Nair AJ, Veena G, Vinayak A (2021) Comparative study of twitter sentiment on covid - 19 tweets. In: 2021 5th International conference on computing methodologies and communication (ICCMC), pp. 1773–1778. https://doi.org/10.1109/ICCMC51019.2021.9418320
17. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
18. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692
19. Tan KL, Lee CP, Anbananthen KSM, Lim KM (2022) Roberta-lstm: A hybrid model for sentiment analysis with transformer and recurrent neural network. IEEE Access 10:21517–21525
20. Monika R, Deivalakshmi S, Janet B (2019) Sentiment analysis of us air-lines tweets using lstm/rnn. In: 2019 IEEE 9th International conference on advanced computing (IACC), pp. 92–95. https://doi.org/10.1109/IACC48062.2019.8971592

21. SivaSai JG, Srinivasu PN, Sindhuri MN, Rohitha K, Deepika S (2020) An automated segmentation of brain mr image through fuzzy recurrent neural network. In: Bio-inspired neurocomputing, pp. 163–179. Springer, ???
22. Bhuvan MS, Rao VD, Jain S, Ashwin T, Guddeti RMR (2015) Semantic sentiment analysis using context specific grammar. In: International conference on computing, communication & automation, pp. 28–35. IEEE
23. Horne L, Matti M, Pourjafar P, Wang Z (2020) Grubert: A gru-based method to fuse bert hidden layers for twitter sentiment analysis. In: Proceedings of the 1st conference of the Asia-Pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing: Student research workshop, pp. 130–138
24. Katz G, Ofek N, Shapira B (2015) Consent: Context-based sentiment analysis. Knowl Based Syst 84:162–178
25. Tang D, Wei F, Qin B, Yang N, Liu T, Zhou M (2016) Sentiment embeddings with applications to sentiment analysis. IEEE Trans Knowl Data Eng 28(2):496–509. https://doi.org/10.1109/TKDE.2015.2489653
26. Vimali J, Murugan S (2021) A text based sentiment analysis model using bi-directional lstm networks. In: 2021 6th International conference on communication and electronics systems (ICCES), pp. 1652–1658. IEEE
27. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30
28. Qiu X, Sun T, Xu Y, Shao Y, Dai N, Huang X (2020) Pre-trained models for natural language processing: A survey. Sci China Technol Sci 63(10):1872–1897
29. Sennrich R, Haddow B, Birch A (2015) Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909
30. Lai S, Yu Z, Wang H (2020) Text sentiment support phrases extraction based on roberta. In: 2020 2nd International conference on applied machine learning (ICAML), pp. 232–237. https://doi.org/10.1109/ICAML51583.2020.00056
31. Thukral S, Kovac S, Paturu M (2023) Chapter 29 - t-test. In: Eltorai AEM., Liu T, Chand R, Kalva SP (eds.) Translational interventional radiology. Handbook for designing and conducting clinical, pp. 139–143. Academic Press, ???. https://doi.org/10.1016/B978-0-12-823026-8.00104-8. https://www.sciencedirect.com/science/article/pii/B9780128230268001048

Authors and Affiliations

Ramalingaswamy Cheruku[1] · Khaja Hussain[1] · Ilaiah Kavati[1] · A. Mallikarjuna Reddy[2] · K. Sudheer Reddy[2]

Khaja Hussain
khajahussain528@gmail.com

Ilaiah Kavati
ilaiahkavati@nitw.ac.in

A. Mallikarjuna Reddy
mallikarjunreddycse@cvsr.ac.in

K. Sudheer Reddy
sudheercse@gmail.com

[1] Department of Computer Science and Engineering, National Institute of Technology Warangal, Warangal 506004, Telangana, India

2 Anurag University, Hyderabad, Telangana, India

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com