applied sciences

MDPI

Article

RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis

Kian Long Tan, Chin Poo Lee * and Kian Ming Lim

Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia
* Correspondence: cplee@mmu.edu.my

Abstract: This paper proposes a novel hybrid model for sentiment analysis. The model leverages the strengths of both the Transformer model, represented by the Robustly Optimized BERT Pretraining Approach (RoBERTa), and the Recurrent Neural Network, represented by Gated Recurrent Units (GRU). The RoBERTa model provides the capability to project the texts into a discriminative embedding space through its attention mechanism, while the GRU model captures the long-range dependencies of the embedding and addresses the vanishing gradients problem. To overcome the challenge of imbalanced datasets in sentiment analysis, this paper also proposes the use of data augmentation with word embeddings by over-sampling the minority classes. This enhances the representation capacity of the model, making it more robust and accurate in handling the sentiment classification task. The proposed RoBERTa-GRU model was evaluated on three widely used sentiment analysis datasets: IMDb, Sentiment140, and Twitter US Airline Sentiment. The results show that the model achieved an accuracy of 94.63% on IMDb, 89.59% on Sentiment140, and 91.52% on Twitter US Airline Sentiment. These results demonstrate the effectiveness of the proposed RoBERTa-GRU hybrid model in sentiment analysis.

Keywords: sentiment analysis; deep learning; Transformer; RoBERTa; GRU

Citation: Tan, K.L.; Lee, C.P.; Lim, K.M. RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis. *Appl. Sci.* **2023**, *13*, 3915. https://doi.org/10.3390/app13063915

Academic Editor: Pengjie Ren

Received: 11 February 2023
Revised: 3 March 2023
Accepted: 6 March 2023
Published: 19 March 2023

1. Introduction

The field of sentiment analysis has seen a tremendous growth in recent years, owing to the increasing use of social media platforms, where people regularly post their opinions and feelings. Sentiment analysis is a text analytics application that identifies the polarity of a given body of text, including positive, negative, and neutral sentiments. By predicting the public voice regarding a specific subject, sentiment analysis acts as an indicator of public preferences, particularly from the economical and political perspectives. For example, businesses can leverage sentiment analysis to understand the likes and dislikes of their customers for their products or services and customize their marketing strategies accordingly, which can ultimately lead to business growth.

Sentiment analysis algorithms range from traditional machine learning methods, such as Naive Bayes, Decision Trees, and Support Vector Machines, to deep learning models, such as Recurrent Neural Networks (RNNs) [1]. RNNs suffer from the vanishing gradients problems when processing long sequences of data, such as text. The vanishing gradient problem occurs when the gradients that are used to update the parameters of the RNN during training become extremely small as they propagate backward through time. This can result in slow or ineffective learning, as the updates to the parameters become negligible and the RNN fails to capture long-term dependencies in the data. This can occur in text when two words or phrases that are semantically related occur far apart in the sequence, making it difficult for the RNN to capture their relationship and incorporate it into the model's predictions. To address these issues, the LSTM and GRU models were proposed. Both models incorporate gating mechanisms that allow the network to selectively remember or forget information from past inputs, thereby enabling the model to capture long-range

dependencies more effectively. In particular, the LSTM model includes a memory cell that can maintain information over long time periods, while the GRU model has a simpler gating mechanism that can be more computationally efficient.

Recently, the Transformer model, which leverages the attention mechanism, has shown excellent performance in natural language processing. The attention mechanism [2] allows the model to selectively weigh different parts of the input sequence to create informative embeddings. Specifically, the attention mechanism computes a weighted sum of the input embeddings, where the weights are determined by a learned compatibility function between the query and the key embeddings. This enables the model to effectively capture long-range dependencies in the input sequence and create more informative representations of the input. In light of this, a hybrid model that combines the strengths of the Transformer and GRU models is proposed in this paper. The hybrid model uses the Robustly Optimized BERT Pretraining Approach (RoBERTa) Transformer model as the encoder, tokenizing the input sequence and encoding it into a discriminative word embedding. The word embedding is then passed into the GRU, capturing the long-range dependencies of the word embedding. A dense layer is incorporated to learn the relationships between the GRU output and the class labels, followed by a classification layer with a softmax function that estimates the probability distributions of the classes. Our main contributions are:

- A RoBERTa-GRU hybrid model that takes advantage of the best features of both the Transformer and GRU models. The RoBERTa model is applied to generate representative word embeddings that capture the unique characteristics of the text. Dynamic attention masking is used in the RoBERTa model to enable the model to learn from a more diverse set of input sequences.
- A GRU model that is utilized to effectively capture long-range dependencies, which are critical for text analytics. The GRU model incorporates update and reset gates to retain long-term memory and mitigate the vanishing gradient problem.
- To tackle the challenge of imbalanced datasets, the paper employs data augmentation with word embeddings. This technique synthesizes additional samples for the minority classes, leading to improved model performance and generalization ability.

2. Related Works

This section reviews the state-of-the-art approaches for sentiment analysis. The existing methods can be broadly divided into two categories: machine learning and deep learning.

2.1. Machine Learning Approaches

Hemakala and Santhoshkumar (2018) [3] conducted sentiment analysis on a dataset collected from Indian Airlines using seven classical machine learning algorithms, namely Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbors, Logistic Regression, Gaussian Naive Bayes, and AdaBoost. The dataset was preprocessed to remove stop words and perform lemmatization. The results showed that the AdaBoost model achieved the highest precision of 84.5%.

Makhmudah et al. (2019) [4] performed sentiment analysis on a dataset of tweets regarding homosexuality in Indonesia using Support Vector Machines (SVM). The dataset was labeled into two classes: positive and negative. The raw data were preprocessed to remove stop words and perform lemmatization and stemming. The Term Frequency–Inverse Document Frequency (TF-IDF) was used as the feature representation for the SVM. The method achieved an accuracy of 99.5% on the dataset.

In the research carried out by Alsalman (2020) [5], Multinomial Naive Bayes was used for sentiment analysis on Arabic tweets. The authors employed a 4-g tokenization technique and Khoja stemmer for text preprocessing, and represented the processed text as TF-IDF features. The Multinomial Naive Bayes model was trained on the dataset containing 2000 tweets, which were labeled into positive and negative classes and evaluated using five-fold cross validation. The proposed approach demonstrated an impressive accuracy of 87.5% on the Arabic tweet dataset.

In the study by Tariyal et al. (2018) [6], various classification algorithms were compared for sentiment analysis on product review tweets. The methods explored included Linear Discriminant Analysis, K-Nearest Neighbors, Classification Furthermore, Regression Trees (CART), SVM, Random Forest, and C5.0. The dataset comprised 1150 tweets that underwent preprocessing steps, including stop words and punctuation removal, case folding, and stemming. The cleaned text was transformed into a Term Document matrix and fed into the classification algorithms for sentiment analysis. The experimental results indicated that the CART method achieved the highest accuracy of 88.99%.

Gupta et al. (2019) [7] proposed a sentiment analysis approach that leverages four different machine learning algorithms: Logistic Regression, Decision Tree, Support Vector Machine, and Neural Network. The sentiment140 dataset was used in the experiments. The raw data were preprocessed, including stop-words removal and lemmatization, and then represented as TF-IDF features. The authors found that the Neural Network model achieved the highest accuracy of 80% compared with the other models.

Similarly, Jemai et al. (2021) [8] developed a sentiment analyzer that employs a set of machine learning algorithms, including Naive Bayes, Bernoulli Naive Bayes, Multinomial Naive Bayes, Logistic Regression, and Linear Support Vector Classification. The experiments were conducted using the "twitter samples" corpus in the Natural Language Toolkit, which includes 5 k positive and 5 k negative tweets. The preprocessing steps included tokenization, stop-words removal, URLs removal, symbols removal, case folding, and lemmatization. The results showed that the Naive Bayes model achieved the highest accuracy of 99.73%.

This subsection discusses different studies that use various machine learning algorithms for sentiment analysis on different datasets, including Indian Airlines feedback, Indonesian tweets about homosexuality, Arabic tweets, product review tweets, and the sentiment140 dataset. Preprocessing steps, such as stop-words removal, lemmatization, and stemming, were employed in the majority of the studies. The algorithms used include AdaBoost, Support Vector Machine, Multinomial Naive Bayes, Logistic Regression, Decision Tree, Bernoulli Naive Bayes, and Linear Support Vector Classification. The results show that the accuracy of the algorithms varies depending on the dataset and the algorithm used, with accuracy ranging from 80% to 99.73%.

2.2. Deep Learning Approaches

In the work by Ramadhani and Goo (2017) [9], a deep learning method, Multilayer Perceptron (MLP), was utilized for sentiment analysis. The authors employed a self-collected dataset of 4000 tweets in Korean and English for their experiments. To preprocess the dataset, several steps were applied, including tokenization, case folding, stemming, and the removal of numbers, stop words, and punctuations. The MLP model consisted of three hidden layers and used Stochastic Gradient Descent (SGD) as the optimizer. The proposed method achieved an accuracy of 75.03%.

Similarly, in the work by Demirci et al. (2019) [10], an MLP model was used for sentiment analysis on Turkish tweets. The authors used a dataset of 3000 positive and negative Turkish tweets with the hashtag "15Temmuz". The data was preprocessed with the Turkish Deasciifier, tokenization, stop-words and punctuation removal, and stemming. To convert the text into embeddings, the authors employed the Word2vec pretrained model. An MLP model, consisting of six dense layers and three dropout layers, was then used for sentiment classification. The proposed method recorded an accuracy of 81.86% on the dataset.

Additionally, Raza et al. (2021) [11] utilized a Multilayer Perceptron (MLP) architecture for sentiment analysis on COVID-19-related tweets. The collected dataset consisted of 101,435 tweets labeled as positive or negative. The preprocessing of the dataset involved the removal of HTML tags and non-letters, tokenization, and stemming. The cleaned texts were then transformed into numerical features using Count Vectorizer and TF-IDF Vectorizer. The resulting features were fed into an MLP model consisting of five hidden layers for

sentiment classification. The study found that the MLP model with Count Vectorizer achieved an accuracy of 93.73%.

In another work, Rhanoui et al. (2019) [12] proposed a hybrid model that combines a Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (Bi-LSTM) for sentiment analysis. The dataset used in their study included 2003 articles and international news in French, which were labeled as neutral, positive, or negative. The texts were represented as embeddings using the pretrained doc2vec model. The hybrid model consists of a convolutional layer, a max pooling layer, a Bi-LSTM layer, a dropout layer, and a classification layer. The results showed that the proposed hybrid model achieved an accuracy of 90.66% on the dataset.

Similarly, Tyagi et al. (2020) [13] utilized a hybrid architecture that integrates CNN and Bi-LSTM for sentiment analysis, utilizing the Sentiment140 dataset. This dataset consists of 1.6 million positive and negative tweets, which were preprocessed to remove stop words, numbers, URLs, Twitter users' names, and punctuations, and subjected to case folding and stemming. The hybrid model consisted of an embedding layer using the GloVe pretrained model, one-dimensional CNN layer, Bi-LSTM layer, multiple fully connected layers, dropout layers, and a classification layer. The proposed hybrid model achieved an accuracy of 81.20% on the Sentiment140 dataset.

Jang et al. (2020) [14] further improved the hybrid architecture of a CNN and Bi-LSTM by incorporating an attention mechanism. The study used the Internet Movie Database (IMDb) dataset with 50k positive and negative reviews for experiments. The texts were represented using the word2vec pretrained embedding model. The model was optimized using the Adam optimizer, L2 regularization, and dropout techniques. The proposed model recorded an accuracy of 90.26% on the IMDb dataset.

In the study by Hossain et al. (2020) [15], a hybrid model combining a CNN and LSTM was proposed for sentiment analysis. The authors used a self-collected dataset containing 100 restaurant reviews from the Foodpanda and Shohoz Food apps, which underwent preprocessing to remove unimportant words and symbols. The texts were then transformed into word embeddings using the word2vec algorithm. The hybrid model consisted of an embedding layer using the pretrained word2vec model, a convolutional layer, a max pooling layer, an LSTM layer, a dropout layer, and a classification layer. The model achieved an accuracy of 75.01% on the self-collected dataset.

In Yang (2018) [16], the author proposed a Recurrent Neural Filter-based Convolutional Neural Network (RNN-CNN) and LSTM model for sentiment analysis. In this model, the RNN was utilized as the convolutional filter. The experiments were conducted using the Stanford Sentiment Treebank dataset, which was transformed into word embeddings using the GloVe word embedding model. The model consisted of an embedding layer using the pretrained GloVe model, a pooling layer, and an LSTM layer. The Adam optimizer and early stopping were used to prevent overfitting. The proposed RNN-CNN-LSTM model achieved an accuracy of 53.4% on the Stanford Sentiment Treebank dataset.

The study conducted by Harjule et al. (2020) [17] aimed to compare the performance of both machine learning and deep learning methods in sentiment analysis. To conduct the experiments, the authors utilized two datasets, namely the Sentiment140 and Twitter US Airline Sentiment datasets. Before the analysis, the datasets were preprocessed to remove noise, such as stop words, URLs, hashtags, punctuations, etc., and tokenized to make the analysis easier. Five methods were used in the comparison, including Multinomial Naive Bayes, Logistic Regression, Support Vector Machine, Long Short-Term Memory, and an ensemble of Multinomial Naive Bayes, Logistic Regression, and Support Vector Machine using majority voting. The results showed that Long Short-Term Memory achieved the highest accuracy of 82% on the Sentiment140 dataset. On the other hand, Support Vector Machine recorded the highest accuracy of 68.9% on the Twitter US Airline Sentiment dataset.

Various studies that employed deep learning approaches for sentiment analysis are presented in this subsection. Most studies used MLP architecture and a hybrid model comprising CNNs and LSTM or Bi-LSTM. Preprocessing the dataset included tokenization,

case folding, stemming, and the removal of numbers, stop words, punctuations, and other irrelevant symbols. The datasets used in the experiments were in different languages such as Turkish, French, English, and Korean, and included tweets, restaurant reviews, news articles, and movie reviews. The accuracy of the models ranged from 53.4% to 93.73%. Some studies also utilized pretrained models for text embeddings, such as Word2vec, GloVe, and Doc2vec, and while text embedding is a crucial component of sentiment analysis, it is surprising that only a few studies utilize Transformer models for this task. The benefits of using Transformers for text embedding include their ability to capture contextual relationships, handle long-range dependencies, and produce highly expressive representations that outperform traditional embedding techniques. A summary of the related works is presented in Table 1.

Table 1. The summary of the related works

Source	Method	Dataset
Hemakala and Santhoshkumar (2018) [3]	DT, RF, SVM, KNN, LR, GNB, AdaBoost	Indian Airlines
Makhmudah et al. (2019) [4]	SVM	Homosexual Tweets
Alsalman (2020) [5]	MNB	Arabic Tweets
Tariyal et al. (2018) [6]	LDA, CART, KNN, SVM, RF, C5.0	Product Reviews
Gupta et al. (2019) [7]	DT, LR, SVM, Neural Network	Sentiment140
Jemai et al. (2021)[8]	NB, MNB, BNB, LR, Linear SVC	NLTK 'twitter_samples'
Ramadhani and Goo (2017) [9]	MLP	Korean and English Tweets
Demirci et al. (2019) [10]	MLP	Turkish Tweets
Raza et al. (2021) [11]	MLP	COVID-19 Tweets
Rhanoui et al. (2019) [12]	CNN-Bi-LSTM	French articles and international news
Tyagi et al. (2020) [13]	CNN-Bi-LSTM	Sentiment140
Jang et al. (2020) [14]	CNN-Bi-LSTM	IMDb
Hossain et al. (2020) [15]	CNN-LSTM	Foodpanda and Shohoz reviews
Yang (2018) [16]	CNN-RNF-LSTM	Stanford Sentiment Treebank
Harjule et al. (2020) [17]	MNB, LR, SVM, LSTM, Ensemble (MNB + LR + SVM)	Sentiment140, Twitter US Airline Sentiment

3. Sentiment Analysis with RoBERTa-GRU

The RoBERTa-GRU hybrid model is designed to tackle the challenges faced in sentiment analysis, especially in imbalanced datasets. The data preprocessing step is crucial in ensuring the accuracy of the sentiment analysis. The preprocessing step involves cleaning the raw text data by removing irrelevant information, such as stop words, punctuations, URLs, and hashtags. The data augmentation technique is then applied to increase the representation of minority classes in the imbalanced dataset, which can lead to better performance in sentiment analysis.

The RoBERTa model serves as the encoder in the hybrid model, where it tokenizes the input text and generates a discriminative word embedding for each token. The GRU component of the hybrid model takes in the word embeddings generated by the RoBERTa encoder and captures the long-range dependencies in the sequence of word embeddings. A dense layer is then incorporated to learn the relations between the GRU outputs and the class labels. Finally, a classification layer with softmax activation is used to estimate the probability distributions of the different sentiment classes.

Overall, the RoBERTa-GRU hybrid model leverages the strengths of both the RoBERTa and GRU models to perform robust sentiment analysis on imbalanced datasets. The process flow of the sentiment analysis with the RoBERTa-GRU model is depicted in Figure 1.

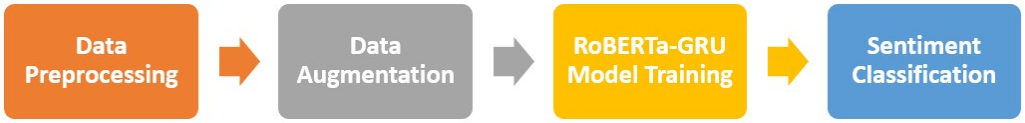

Figure 1. The process flow of the sentiment analysis with RoBERTa-GRU.

3.1. Data Preprocessing

Data preprocessing is a crucial step in sentiment analysis as it ensures that the input text is in a standardized format that can be effectively processed by the machine learning algorithms. The preprocessing steps applied in the proposed system are aimed at cleaning the raw text and removing any elements that may negatively impact the sentiment analysis results.

Case folding is an important preprocessing step that ensures that the text is standardized into a consistent case. This helps to prevent case-sensitive issues that may arise due to the inconsistent use of upper- and lower-case letters in the text. As the texts used in the analysis are mostly tweets, the removal of numbers, punctuation, and special symbols is carried out to eliminate any irrelevant information from the text.

Stop-words removal is another important preprocessing step as stop words bring less meaning in sentiment analysis. Stop words, such as "a", "an", "the", etc., act as functional words but do not contribute much to the sentiment of the text. By removing these words, the focus is shifted towards the content words that carry more meaning and relevance to the sentiment analysis task.

Overall, the data preprocessing steps applied in the proposed system help to improve the efficiency and accuracy of the sentiment analysis by removing any irrelevant information from the raw text and standardizing the text into a format that can be effectively processed by the machine learning algorithms.

3.2. Data Augmentation

To overcome the issue of deep learning models requiring a large number of samples for effective model learning, data augmentation is leveraged to increase the number of samples in the dataset. A larger sample size contributes to better model learning and enhances the performance and generalization ability of the model. However, data augmentation in the field of Natural Language Processing (NLP) can be challenging due to the complexity of human language.

There are several popular techniques for text data augmentation, including Thesaurus Substitution [18], Text Generation [19], and Word Embedding [20]. The Thesaurus Substitution technique involves replacing words or phrases with their synonyms to generate new samples. Text Generation, on the other hand, produces entirely new sentences based on the original sentences, and Word Embedding augmentation substitutes words by identifying the embedding with the highest cosine similarity through a K-Nearest Neighbor. Some commonly used pretrained word embeddings include word2vec, FastText, and Global Vectors for Word Representation (GloVe).

In this study, the GloVe word embedding technique was chosen for data augmentation. The data augmentation was applied to the imbalanced Twitter US Airline dataset to generate additional samples for the minority classes and balance the class distributions.

3.3. RoBERTa-GRU

This subsection presents the architecture of the proposed RoBERTa-GRU model, as depicted in Figure 2.

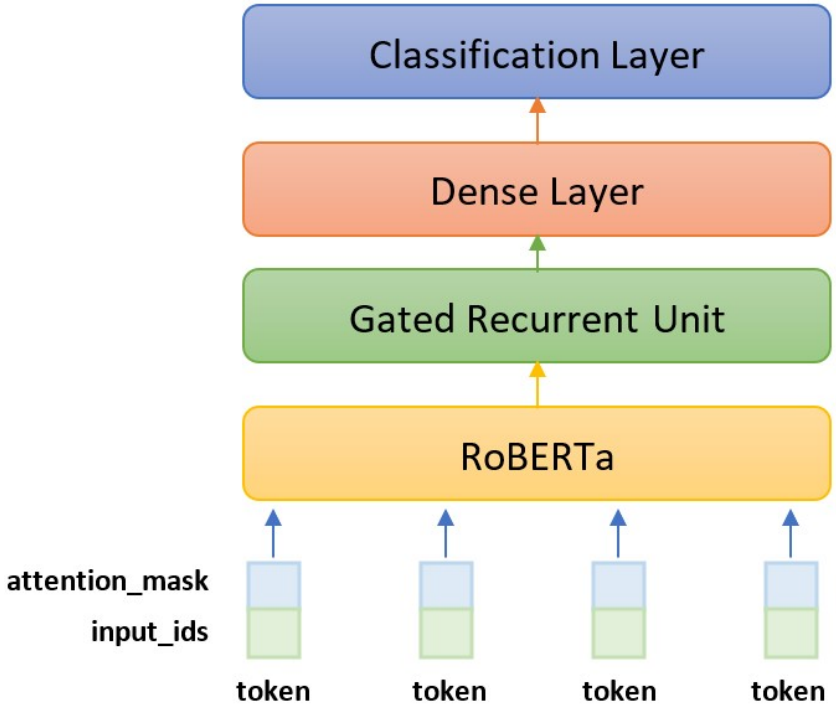

Figure 2. The architecture of the proposed RoBERTa-GRU model.

3.3.1. RoBERTa

The proposed model employs a Robustly Optimized BERT Pretraining Approach (RoBERTa) as its first layer. The RoBERTa is an improved version of the pretrained Bidirectional Encoder Representations from Transformers (BERT) model. Both the BERT and RoBERTa are based on the Transformer architecture.

The Transformer model was designed for sequence-to-sequence tasks where long-range dependencies are important. It uses self-attention mechanisms instead of recurrence or convolution to identify dependencies between inputs and outputs. The self-attention mechanism assigns higher weights to relevant inputs, thus reducing the length of the sequence.

The Transformer consists of two main components: the encoder and the decoder. The encoder is made up of a self-attention layer and a feed-forward network, and the decoder consists of a self-attention layer, an encoder–decoder attention layer, and a feed-forward network. The encoder is responsible for reading the input text, while the decoder focuses on making predictions. In the proposed model, only the encoder part of the RoBERTa is used as the text encoder.

BERT was created to overcome the limitations of unidirectional methods that limit context learning. It was originally used for two tasks: Masked Language Modeling and Next Sentence Prediction. In Masked Language Modeling, the BERT uses masked token features to predict missing words in context, while in Next Sentence Prediction, the BERT uses semantic meaning from different text segments or documents to predict the next sentence.

The RoBERTa has some advantages over BERT. For instance, RoBERTa uses byte-level Byte Pair Encoding for tokenization, resulting in a smaller vocabulary and less computational resources compared with BERT's character-level Byte Pair Encoding. Additionally, BERT uses static masking where the masking is performed once during data preprocessing, while RoBERTa uses dynamic masking, where the input sequence is duplicated and different attention masks are applied, enabling the RoBERTa model to learn from different input sequences. The RoBERTa is also trained on much larger datasets using a larger batch size and longer sequence for a longer time. It was trained on four datasets: BookCorpus + English Wikipedia (16 GB), CC-News (76 GB), OpenWebText (38 GB), and Stories (31 GB).

To prepare the input data for the RoBERTa model, the pretrained RoBERTa tokenizer is employed in this study. The tokenizer breaks down the raw text into a sequence of subword tokens, leveraging the large corpus of text on which the RoBERTa was trained to learn a robust vocabulary. This process helps preserve the semantic meaning of the text while minimizing the impact of out-of-vocabulary (OOV) words. Following tokenization, each token is assigned a unique input ID, representing its index in the RoBERTa vocabulary. An attention mask is also assigned to each token, indicating its relevance in the context of the input sequence. This mechanism enables the model to focus on the most salient tokens while disregarding those that are less important, thus improving the model's performance on downstream tasks. The resulting input IDs and attention masks are then fed into the RoBERTa model comprising 12 layers, each with 768 hidden states. The RoBERTa model processes the input sequence in a self-attentive manner, capturing contextual information at multiple levels of abstraction. To further refine the output of the RoBERTa model, a GRU is employed as a feature extractor. By incorporating the GRU, the model can leverage both the contextual information captured by the RoBERTa and the long-range dependencies between tokens to generate more accurate predictions.

3.3.2. Gated Recurrent Unit

The Gated Recurrent Unit (GRU) is a type of Recurrent Neural Network (RNN) that is designed to capture long-range dependencies in the encoding from the RoBERTa model. The GRU was introduced to address the vanishing gradient problem that is common in RNNs. The GRU has two main components, namely the update gate and the reset gate. The update gate determines which information to preserve, while the reset gate controls how much past information should be forgotten.

At time step t, the input x_t is processed to produce the output of the update gate z_t, reset gate r_t, memory cell c_t, and hidden state h_t. The calculation of these variables can be represented as follows:

$$\begin{aligned} z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) \\ r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \\ c_t &= \tanh(W_c x_t + U_c(r_t * h_{t-1}) + b_c) \\ h_t &= z_t * h_{t-1} + (1 - z_t) * c_t \end{aligned} \tag{1}$$

In the above equations, σ is the sigmoid function and $*$ is the element-wise multiplication operation. The weights to be multiplied with the input and hidden states h_{t-1} are represented as (W_z, W_r, W_c) and (U_z, U_r, U_c), respectively, where the subscript z, r, and c denote the update gate, reset gate, and memory cell, respectively. The biases of the update gate, reset gate, and memory cell are denoted as (b_z, b_r, b_c).

The sigmoid function ensures that the output values of the update gate and reset gate are within the range of [0, 1]. The memory cell utilizes the reset gate to decide whether to keep the past information. When the value of the reset gate is close to zero, most of the past information is discarded.

The hidden state combines the information from the past hidden state and the current memory cell weighted by the update gate. If the value of the update gate is close to zero, then most of the past information is ignored, and the hidden state is updated with the current memory cell content. Finally, the output of the GRU layer is passed to the flatten layer.

3.3.3. Flatten Layer

The flatten layer plays a crucial role in the neural network architecture of the RoBERTa-GRU model. It is placed between the GRU layer and the subsequent dense layer. The flatten layer performs the operation of reshaping the output of the GRU layer, so that it can be fed into the dense layer. The output of the GRU layer is a tensor with multiple dimensions,

while the dense layer accepts a tensor with a flat shape. The flatten layer transforms the multi-dimensional tensor into a one-dimensional tensor by unrolling all its elements.

3.3.4. Dense Layer

The dense layer, also known as the fully connected layer, is a key component in the RoBERTa-GRU model. It connects all the neurons in the previous layer to all the neurons in the next layer, creating a dense connectivity pattern. In the RoBERTa-GRU model, there are two dense layers that are used.

The first dense layer is responsible for capturing the relationship between the hidden states generated by the GRU layer and the class labels. It receives the flattened output from the GRU layer and applies a series of matrix operations to model the interactions between the hidden states and class labels.

The second dense layer performs the final classification by producing the probability distributions of the classes. This layer uses the softmax activation function, which transforms the output values into a probability distribution by squashing the output values within the range of [0, 1] and ensuring that the sum of the probabilities is equal to 1. The softmax function allows the model to output the probabilities of different classes, which can be used to determine the most likely class given the input.

3.3.5. Model Training Parameters

The RoBERTa-GRU model is trained with a categorical cross-entropy loss function, which is a common loss function for multiclass classification problems. The categorical cross-entropy loss measures the dissimilarity between the predicted probability distribution and the true label distribution. Let the predicted probability distribution for the i-th sample as $\hat{y}_i$ and the true label distribution as y_i. Then, the categorical cross-entropy loss function can be defined as:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \log \hat{y}_i \tag{2}$$

where N is the number of samples in the dataset.

The objective of the optimization process is to minimize the categorical cross-entropy loss function by updating the model parameters θ. The RoBERTa-GRU model is optimized using the Nesterov-accelerated Adaptive Moment Estimation (Nadam) optimizer, which combines the advantages of the Nesterov momentum and the Adaptive Moment Estimation (Adam) optimization methods. The Nesterov momentum accelerates the convergence of the optimization process by incorporating the gradient information from the future. The Nadam optimization method updates the model parameters as:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\ n_t &= \beta_2 n_{t-1} + (1-\beta_2) g_t^2 \\ \hat{m}_t &= m_t / (1-\beta_1^t) \\ \hat{n}t &= n_t / (1-\beta_2^t) \\ \theta t + 1 &= \theta_t - \frac{\alpha}{\sqrt{\hat{n}_t} + \epsilon} \hat{m}_t \end{aligned} \tag{3}$$

where g_t is the gradient of the loss function with respect to the model parameters at time step t, m_t and n_t are the first and second moment estimates, respectively, β_1 and β_2 are the exponential decay rates for the first and second moment estimates, respectively, α is the learning rate, $\hat{m}_t$ and $\hat{n}_t$ are the bias-corrected first and second moment estimates, respectively, and ϵ is a small constant added to the denominator to avoid division by zero. The Nesterov-accelerated Adaptive Moment Estimation optimizer updates the model parameters in such a way that they converge faster to the optimal solution compared with traditional optimization methods.

4. Dataset

The proposed RoBERTa-GRU model was evaluated on three publicly available sentiment analysis datasets: the Internet Movie Database (IMDb) dataset, the Sentiment140 dataset, and the Twitter US Airline dataset.

The IMDb dataset [21] contains a total of 50,000 movie reviews, evenly divided between 25,000 positive and 25,000 negative reviews. This makes it an ideal dataset for evaluating the performance of binary classification models, as it has a well-balanced distribution of classes.

The Sentiment140 dataset [22], released by Stanford University, is a widely-used benchmark for sentiment analysis algorithms. The dataset consists of 1.6 million tweets, each annotated as either positive or negative. The distribution of sentiment labels is well balanced, with 50% of the tweets being positive and 50% being negative, making it an ideal dataset for evaluating binary classification models. The tweets in the Sentiment140 dataset are related to products, brands, or topics, and provide valuable insights into the public sentiment towards different entities.

The Twitter US Airline dataset, on the other hand, contains 14,160 tweets related to American Airlines and is an imbalanced dataset, with 9178 negative tweets, 2363 positive tweets, and 3099 neutral tweets. To mitigate the skewness in the sample distribution, data augmentation using GloVe word embeddings was applied to oversample the minority classes. As shown in Figure 3, after data augmentation, the classes in the Twitter US Airline dataset have an even number of samples.

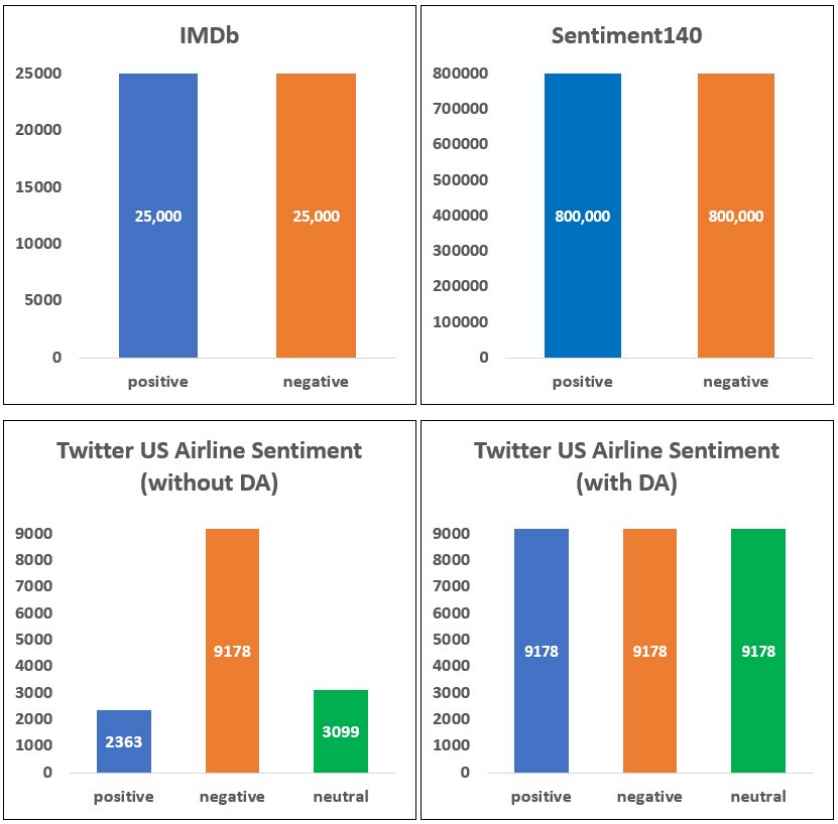

Figure 3. The sample distributions of the datasets.

5. Hyperparameter Tuning

The hyperparameters of the proposed RoBERTa-GRU model were carefully selected to optimize its performance. To determine the optimal hyperparameters, grid search was conducted on the smallest dataset, i.e., the Twitter US Airline dataset after data augmentation. The grid search involved evaluating the model performance for different

combinations of hyperparameters and selecting the combination that results in the highest accuracy. The hyperparameters and their tested values are summarized in Table 2.

Table 2. The hyperparameter tuning of RoBERTa-GRU model.

Hyperparameter	Tested Values	Optimal Value
GRU	64, 128, 256, 512	256
Optimizer	Adam, Nadam, SGD	Nadam
Learning rate	0.00001, 0.0001, 0.001	0.00001

Table 3 presents the results of the hyperparameter tuning for the number of GRU in the RoBERTa-GRU model. The number of GRUs plays a crucial role in determining the capacity of the model to capture the long-range dependencies of the RoBERTa embedding. By increasing the number of GRUs, the model can learn more complex relationships between the input and output. However, using a large number of GRUs might also lead to overfitting, especially when the training data is limited. The results in Table 3 suggest that the optimal number of GRUs is 256. At this number of units, the RoBERTa-GRU model achieved the highest accuracy in the sentiment classification task. This indicates that a hidden layer with 256 GRUs is capable of capturing the complex relationships between the input and output while avoiding overfitting.

Table 3. The experimental results of different GRUs (optimizer = Nadam, learning rate = 0.00001).

GRU	Accuracy (%)
64	91.07
128	91.41
256	91.52
512	90.83

The grid search for different optimizers was an important step in determining the optimal hyperparameters for the RoBERTa-GRU model. The results in Table 4 indicate that the Nadam optimizer is the most effective optimizer for the RoBERTa-GRU model as it not only yields the highest accuracy but also allows for the quickest convergence during the training process. By choosing the Nadam optimizer, the model can effectively learn the underlying relationships in the data, leading to better performance in sentiment analysis tasks.

Table 4. The experimental results of different optimizers (GRU = 256, learning rate = 0.00001).

Optimizer	Accuracy (%)	Execution Time (s)
Adam	90.72	1821.97
Nadam	91.52	1584.04
SGD	84.66	4520.34

The results of the grid search for the learning rate are shown in Table 5. The learning rate, as a hyperparameter, has a crucial role in determining the convergence speed and the final performance of the model. The experiment results indicate that the optimal value for the learning rate was found to be 0.00001. This value balances the trade-off between underfitting and overfitting. If the learning rate is too high, the model may underfit, meaning that it will not be able to capture the underlying patterns in the data and may result in poor performance. On the other hand, if the learning rate is too low, the model may overfit, which means that it will learn the noise in the data and may perform poorly on unseen data. The learning rate of 0.00001 provides a good balance between these two extremes and results in the best performance for the RoBERTa-GRU model.

Table 5. The experimental results of different learning rates (GRU = 256, optimizer = Nadam).

Learning Rate	Accuracy (%)
0.00001	91.52
0.0001	89.81
0.001	32.69

6. Experimental Results and Analysis

This section presents the experimental results of the BERT, RoBERTa, BERT-GRU, and RoBERTa-GRU on the Twitter US Airline Sentiment dataset without data augmentation, as shown in Table 6. The table shows that the RoBERTa outperforms BERT, achieving an accuracy of 90.67% compared with 89.36% for BERT. The BERT-GRU model achieved an accuracy of 89.56% on the Twitter US Airline Sentiment dataset, while the RoBERTa-GRU achieved the highest accuracy of 91.28%. These results indicate that the addition of a GRU layer further improved the performance of the RoBERTa. Therefore, based on these results, the RoBERTa-GRU was chosen for the proposed models. These results demonstrate the effectiveness of the RoBERTa and its ability to capture more nuanced features in the text data.

Table 6. The Comparison Results between BERT and RoBERTa.

Model	Accuracy (%)
BERT	89.36
BERT-GRU	89.56
RoBERTa	90.67
RoBERTa-GRU	91.28

The effectiveness of the proposed RoBERTa-GRU model is also analyzed and compared against the state-of-the-art approaches. The datasets used in the experiments are divided into three portions: 60% for training, 20% for validation, and 20% for testing. The model is trained for a maximum of 100 epochs with a batch size of 32. To avoid overfitting, an early stopping technique is employed with a patience of 30 epochs based on the validation accuracy.

Table 7 compares the accuracy, precision, recall, and F1-score of various machine learning and deep learning models on the IMDb dataset. Among the machine learning models, logistic regression achieves the highest accuracy of 87.12%, whereas the GRU model performs the best among the deep learning methods with an accuracy of 87.88%. However, the proposed RoBERTa-GRU model surpasses all other models with an accuracy of 94.63%. The superiority of the RoBERTa-GRU model can be attributed to several factors. Firstly, the RoBERTa model is a pretrained Transformer-based language model that has been fine-tuned on a large corpus of text data. The pretraining allows the model to learn rich and meaningful representations of the input text, making it more effective in capturing the underlying sentiment of the movie reviews. Moreover, the RoBERTa-GRU model combines the strengths of the RoBERTa and GRU models. The RoBERTa-GRU model uses the RoBERTa's pretrained embeddings as input to the GRU network, which allows the model to learn the context and dependencies of the text more effectively. Additionally, the GRU network's ability to handle sequential data enables the model to capture the long-range relationships between words and phrases, which is essential in sentiment analysis.

Table 7. The comparison results on the IMDb.

Methods	Accuracy	Precision	Recall	F1-Score
Naïve Bayes [8]	87.01	87	87	87
Logistic Regression [23]	87.12	90	90	90
Decision Tree [24]	73.46	74	73	73
KNN [23]	77.37	78	77	77
AdaBoost [24]	83.37	83	83	83
GRU [25]	87.88	88	88	88
LSTM [25]	85.11	85	85	85
BiLSTM [26]	86.28	87	86	86
CNN-LSTM [12]	86.07	86	86	86
CNN-BiLSTM [12]	86.16	86	86	86
RoBERTa-GRU	**94.63**	**95**	**95**	**95**

Table 8 presents a comparison of various machine learning and deep learning methods on the Sentiment140 dataset. The proposed RoBERTa-GRU model records the highest accuracy of 89.59% among all the methods. This suggests that the RoBERTa-GRU model has a better capability of predicting the sentiment of a given text than other methods. One reason for the superior performance of the RoBERTa-GRU is that the RoBERTa is pretrained using a much larger corpus and a more diverse set of pretraining tasks than other models, such as the LSTM and GRU approaches. This results in better sentence embedding and contextual understanding, which enables the RoBERTa-GRU to capture more nuanced patterns in the text data and make more accurate predictions. Furthermore, the gating mechanism of the GRU selectively updates the hidden state based on the input and the previous hidden state. GRUs have been observed to outperform LSTMs in cases where the data have short-term dependencies. The Sentiment140 dataset consists of relatively short tweets, making GRUs a fitting choice for this particular task.

Table 8. The comparison results on the Sentiment140 dataset.

Methods	Accuracy	Precision	Recall	F1-Score
Naïve Bayes [8]	76.57	77	77	77
Logistic Regression [23]	78.01	78	78	78
Decision Tree [24]	62.34	69	62	59
KNN [23]	60.39	66	60	57
AdaBoost [24]	69.94	71	70	69
GRU [25]	78.96	78	78	78
LSTM [25]	79.10	79	79	79
BiLSTM [26]	78.53	78	78	78
CNN-LSTM [12]	77.53	77	77	77
CNN-BiLSTM [12]	77.58	77	77	77
RoBERTa-GRU	**89.59**	**90**	**90**	**90**

Table 9 presents a comparison of various sentiment analysis models on the Twitter US Airline dataset. The results indicate that the proposed RoBERTa-GRU model outperforms all the other models, with an accuracy of 91.52% and an F1-score of 91%, which represents an improvement of 11.05% in accuracy and 19.52% in F1-score compared with the logistic regression model that records an accuracy of 80.5% and an F1-score of 72%. The RoBERTa-GRU model uses the RoBERTa Transformer to capture contextual information from the text, and the GRU layer to model the sequential information. This combination of a Transformer and a recurrent layer results in better performance than models that only use one of these layers, such as the RoBERTa-LSTM. In addition to that, the proposed RoBERTa-GRU model benefits from data augmentation using the GloVe word embedding technique, which helps to mitigate the imbalanced dataset problem and synthesize more samples for better model learning. This data augmentation technique enhances the performance of the model, resulting in an improved accuracy of 91.52%.

Table 9. The comparison results on the Twitter US Airline Sentiment.

Methods	Accuracy	Precision	Recall	F1-Score
Naïve Bayes [8]	69.5	79	44	45
Logistic Regression [23]	80.5	78	69	72
Decision Tree [24]	71.14	62	56	58
KNN [23]	68.41	60	60	60
AdaBoost [24]	74.59	67	63	65
GRU [25]	78.55	73	71	72
LSTM [25]	77.56	71	69	69
BiLSTM [26]	77.46	71	69	70
CNN-LSTM [12]	76.02	68	69	69
CNN-BiLSTM [12]	77.32	70	65	67
RoBERTa-LSTM	85.69	86	86	86
RoBERTa-GRU	**91.52**	**91**	**91**	**91**

Overall, the proposed RoBERTa-GRU model outshines other methods on three datasets, reflecting its superior performance in sentiment analysis. The pretrained RoBERTa model has been trained on an enormous corpus of text data, enabling it to learn complex language patterns and relationships between words and phrases. Additionally, the RoBERTa's dynamic masking patterns improve its ability to generalize and adapt to new text sequences. The gating mechanism of a GRU addresses the vanishing gradients issue by selectively updating its hidden state, allowing it to better capture the long-range dependencies in text.

7. Conclusions

This research study presents the novel RoBERTa-GRU model, which combines the state-of-the-art deep learning techniques in NLP, RoBERTa and GRU models. The RoBERTa, which is part of the Transformer family, utilizes dynamic attention masking to produce meaningful text embeddings, thus enabling the model to generalize better. The GRU model, on the other hand, enables the capture of long-term dependencies in text sequences, and its gating mechanism helps to overcome the vanishing gradient problem that is commonly encountered in Recurrent Neural Networks (RNNs). In addition, the proposed model incorporates data augmentation through the use of the GloVe word embedding technique on the imbalanced Twitter US Airline dataset to improve the minority class representation.

The results of the experiments carried out on the IMDb, Sentiment140, and Twitter US Airline datasets indicate that the RoBERTa-GRU model outperforms all other comparison methods, with accuracy scores of 94.63%, 89.59%, and 91.52%, respectively. The combination of the RoBERTa and GRU results in a powerful and efficient model for sentiment analysis, making it a promising solution for a variety of NLP tasks.

Author Contributions: Conceptualization, K.L.T. and C.P.L.; methodology, K.L.T., C.P.L. and K.M.L.; software, K.L.T. and C.P.L.; validation, K.L.T. and C.P.L.; formal analysis, K.L.T.; investigation, K.L.T.; resources, K.L.T.; data curation, K.L.T. and C.P.L.; writing—original draft preparation, K.L.T.; writing—review and editing, C.P.L. and K.M.L.; visualization, K.L.T. and C.P.L.; supervision, C.P.L. and K.M.L.; project administration, C.P.L.; funding acquisition, C.P.L. All authors have read and agreed to the published version of the manuscript.

Funding: The research in this work was supported by the Fundamental Research Grant Scheme of the Ministry of Higher Education under award number FRGS/1/2021/ICT02/MMU/02/4 and Multimedia University Internal Research Grant with award number MMUI/220021.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Birjali, M.; Kasri, M.; Beni-Hssane, A. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowl.-Based Syst.* **2021**, *226*, 107134. [CrossRef]
2. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
3. Hemakala, T.; Santhoshkumar, S. Advanced classification method of twitter data using sentiment analysis for airline service. *Int. J. Comput. Sci. Eng.* **2018**, *6*, 331–335. [CrossRef]
4. Makhmudah, U.; Bukhori, S.; Putra, J.A.; Yudha, B.A.B. Sentiment Analysis Of Indonesian Homosexual Tweets Using Support Vector Machine Method. In Proceedings of the 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), Jember, Indonesia, 16–17 October 2019; pp. 183–186.
5. AlSalman, H. An improved approach for sentiment analysis of arabic tweets in twitter social media. In Proceedings of the 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 19–21 March 2020; pp. 1–4.
6. Tariyal, A.; Goyal, S.; Tantububay, N. Sentiment Analysis of Tweets Using Various Machine Learning Techniques. In Proceedings of the 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), Bhopal, India, 28–29 December 2018; pp. 1–5.
7. Gupta, A.; Singh, A.; Pandita, I.; Parashar, H. Sentiment analysis of Twitter posts using machine learning algorithms. In Proceedings of the 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 13–15 March 2019; pp. 980–983.
8. Jemai, F.; Hayouni, M.; Baccar, S. Sentiment Analysis Using Machine Learning Algorithms. In Proceedings of the 2021 International Wireless Communications and Mobile Computing (IWCMC), Harbin, China, 28 June–2 July 2020; pp. 775–779.
9. Ramadhani, A.M.; Goo, H.S. Twitter sentiment analysis using deep learning methods. In Proceedings of the 2017 7th International Annual Engineering Seminar (InAES), Yogyakarta, Indonesia, 1–2 August 2017; pp. 1–4.
10. Demirci, G.M.; Keskin, Ş.R.; Doğan, G. Sentiment analysis in Turkish with deep learning. In Proceedings of the 2019 IEEE International Conference on Big Data, Los Angeles, CA, USA, 9–12 December 2019; pp. 2215–2221.
11. Raza, G.M.; Butt, Z.S.; Latif, S.; Wahid, A. Sentiment Analysis on COVID Tweets: An Experimental Analysis on the Impact of Count Vectorizer and TF-IDF on Sentiment Predictions using Deep Learning Models. In Proceedings of the 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2), Islamabad, Pakistan, 20–21 May 2021; pp. 1–6.
12. Rhanoui, M.; Mikram, M.; Yousfi, S.; Barzali, S. A CNN-BiLSTM model for document-level sentiment analysis. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 832–847. [CrossRef]
13. Tyagi, V.; Kumar, A.; Das, S. Sentiment Analysis on Twitter Data Using Deep Learning approach. In Proceedings of the 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 18–19 December 2020; pp. 187–190.
14. Jang, B.; Kim, M.; Harerimana, G.; Kang, S.u.; Kim, J.W. Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism. *Appl. Sci.* **2020**, *10*, 5841. [CrossRef]
15. Hossain, N.; Bhuiyan, M.R.; Tumpa, Z.N.; Hossain, S.A. Sentiment analysis of restaurant reviews using combined CNN-LSTM. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; pp. 1–5.
16. Yang, Y. Convolutional neural networks with recurrent neural filters. *arXiv* **2018**, arXiv:1808.09315.
17. Harjule, P.; Gurjar, A.; Seth, H.; Thakur, P. Text classification on Twitter data. In Proceedings of the 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), Jaipur, India, 7–8 February 2020; pp. 160–164.
18. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 649–657.
19. Kafle, K.; Yousefhussien, M.; Kanan, C. Data augmentation for visual question answering. In Proceedings of the 10th International Conference on Natural Language Generation, Santiago de Compostela, Spain, 4–7 September 2017; pp. 198–202.
20. Wang, W.Y.; Yang, D. That is so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2557–2563.
21. Maas, A.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
22. Go, A.; Bhayani, R.; Huang, L. Twitter sentiment classification using distant supervision. *CS224N Proj. Rep. Stanf.* **2009**, *1*, 2009.
23. Dholpuria, T.; Rana, Y.; Agrawal, C. A Sentiment analysis approach through deep learning for a movie review. In Proceedings of the 2018 8th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 24–26 November 2018; pp. 173–181.
24. Vadivukarassi, M.; Puviarasan, N.; Aruna, P. An exploration of airline sentimental tweets with different classification model. *Int. J. Res. Eng. Appl. Manag.* **2018**, *4*, 72–77.

25. Hossen, M.S.; Jony, A.H.; Tabassum, T.; Islam, M.T.; Rahman, M.M.; Khatun, T. Hotel review analysis for the prediction of business using deep learning approach. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 25–27 March 2021; pp. 1489–1494.
26. Vimali, J.; Murugan, S. A Text Based Sentiment Analysis Model using Bi-directional LSTM Networks. In Proceedings of the 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, 8–10 July 2021; pp. 1652–1658.