Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org

Volume 24, Issues 3, pp. 299–314, DOI 10.12694/scpe.v24i3.2122

ASPECT-BASED TEXT CLASSIFICATION FOR SENTIMENTAL ANALYSIS USING ATTENTION MECHANISM WITH RU-BILSTM

SANDEEP YELISETTI* AND NELLORE GEETHANJALI†

Abstract. Sentiment analysis has gained increasing attention from an educational and social perspective with the huge expansion of user interactions due to the Web's significant improvement. The connection between an opinion target's polarity scores and other aspects of the content is defined by aspect-based sentiment analysis. Identifying aspects and determining their different polarities is quite complicated because they are frequently implicit. To overcome these difficulties, efficient hybrid methods are used in aspect-based text classification in sentiment analysis. The existing process evaluates the aspects of polarity by using a Convolutional neural network, and it does not work with Big data. In this work, aspect-based text classification and attention mechanisms are used to assist in filtering out irrelevant information and quickly locating the essential features in big data. Initially, the data is collected, and then the data is preprocessed by using Tokenization, Stop word removal, Stemming, and Lemmatization. After preprocessing, the features are vectorized and extracted using Bag-of-Words and TF-IDF. Then, the extracted features are given into word embeddings by GloVe and Word2vec. It uses Deep Recurrent based Bidirectional Long Short Term Memory (RU-BiLSTM) for aspect-based sentiment analysis. The RU-Bi-LSTM method integrates aspect-based embeddings and an attention mechanism for text classification. The attention mechanism focuses on more crucial aspects and the bidirectional LSTM to maintain context in both ways. Finally, the binary and ternary classification outcomes are obtained using the final dense softmax output layer. The proposed RU-BiLSTM uses four reviews and two Twitter datasets. The results of the studies demonstrate the efficacy of the RU-BiLSTM model, which outperformed aspect-based classifications on lengthy reviews and short tweets in terms of evaluation.

Key words: aspect-based sentiment analysis, attention mechanism, aspect level, word2vec, TF-IDF, Convolutional Neural Network, big data, BiLSTM

1. Introduction. The Sentiment Analysis is useful in finding the feelings expressed in the text's sections. Aspect-Based Sentiment Analysis (ABSA), a pioneering technique, asserts that there are three stages at which SA research is carried out: text, sentence, word, or aspect. SA implies that each document expresses a view about a single entity at the level of the document. It is supposed that the text only covers one subject, yet this is frequently untrue. Every sentence is viewed as a separate entity in a sentencing-level strategy, and it is thought that the sentence should only include one assumption. Two tasks of the sentence-level analysis are grouping estimation and subjective characterization.

The sentiment analysis (SA) of texts aims to gather and examine facts from the private data placed on social media. Nowadays, there is an interest in the area of natural language processing (NLP) related to SA because of its broad range of educational and commercial uses and the expansion of social networks. As a result, various tools and techniques have been recommended to determine the polarity of texts. In most SA applications, polarity recognition is a crucial binary classification problem. Earlier SA methods relied on deep procedures and cleverly engineered effective characteristics to achieve acceptable polarity classification outcomes [11].

To overcome the earlier restrictions, learning word embedding has been recommended in several recent kinds of research [7, 28, 30]. A thick real-valued vector called word embedding, which considers different lexical relationships, is produced by a neural language model [31, 33]. As a result, deep neural networks (DNN) frequently use word embedding as their input in existing NLP works. Experts from a wide range of fields, including computer vision [32], finance [3], medical informatics, and multimedia sentiment analysis [2], have been paying more and more attention to DNNs in recent years.

As e-commerce grows, increasing numbers of individuals are eager to express their views and thoughts on branded items after using them, resulting in an enormous amount of remark messages. These little compositions

*Department of Computer Science Technology,Sri Krishnadevaraya University, Ananthapuram, India (sandeepyelisetti22@outlook.com)

†Department of Computer Science & Technology, Sri Krishnadevaraya University, Ananthapuram, India

frequently have a significant subjective element, occasionally incorporating many psychological characteristics within a single sentence. Short text remarks are also incredibly informal. Because of this, the text's subject is ambiguous, difficult to locate, linguistically inconsistent, and, more concerning, challenging for scholars to use effectively. Nevertheless, most recent studies on analyzing brief remark texts' sentiments use a coarse-grained approach, whereby just one sentiment trend is gleaned from a sentence [9].

A simple positive signal will be given to this remark in coarse-grained sentiment analysis. In actuality, the user's feedback was merely positive regarding the product's look and unfavorable regarding its performance. As it has been shown, coarse-grained sentiment classification is unable to capture the particulars that users care about fully. The emotive inclinations of various features in some customer review ontologies can be identified using aspect-based sentiment analysis. Depending on this benefit, aspect-based sentiment classification of brief user review texts can assist businesses in enhancing specific products while also assisting consumers in making better choices.

This work aimed to use deep learning research techniques to increase the accuracy of the aspect-based Attention mechanism for sentiment analysis in response to the issues mentioned earlier. This approach can record remarks at the aspect level and extract local feature information. Furthermore, it can prevent gradient expansion and disappearing problems, enhance model fidelity, and lower computing overhead. In addition, this approach can be applied practically to the emotional evaluation of brief texts like microblog responses and social attitudes postings. Those are important in relevant domains in all spheres of life.

1.1. Motivation. The motivation for this research is to introduce a new method for aspect-based sentiment analysis that can efficiently handle big data and accurately identify the polarity of implicit aspects in text. The work highlights the challenges of aspect-based sentiment analysis and the limitations of existing methods, which cannot work with big data. The proposed method uses pre-processing, vectorization, and deep learning models like RU-BiLSTM with an attention mechanism to classify text based on aspects and their polarities. The research aims to demonstrate the efficacy of this new method through experiments on four review datasets and two Twitter datasets. This work provides a solution to the challenges of aspect-based sentiment analysis and highlights the importance of accurate sentiment analysis in the context of social and educational applications.

1.2. Objectives of the study. The following objectives are the focus of this research study:

1. This study makes use of LSTM and BiLSTM to analyze two kinds of datasets. They are long reviews and short tweets from social media.
2. By testing with both long and short user evaluations, it explores the capacity of LSTM and BiLSTM as well as the attention process in gathering contextual information.
3. We provide an attention mechanism to effectively improve the emotion polarities of words, identify important data in the text, and collect the phrase within the text that are highly connected to the extended gap and encoded dependency.

1.3. Contributions of this study. The main contribution of the suggested method is given below:

1. The information was first cleaned during the preprocessing stage to remove any special symbols, syntax, English, numerals, etc.
2. The features are vectorized by using Bag-of-Words and TF-IDF.
3. Word2Vec and GloVe are used for extracting features.
4. The attention mechanism effectively increases the sentiment polarity of the words, detects the important data in the text and catches the words within the text which are strongly connected to long space and encoding dependency.
5. Attention mechanism provides specific attention to the data produced from the hidden layers of RU-BiLSTM.

The remaining part of our research article is written as follows: Section 2 discusses the related work on Chinese corpus comments, Attention Mechanism, and Aspect level Sentiment Analysis. Section 3 shows the general working methodology of the proposed work. Section 4 evaluates the implementation and results of the proposed method. Section 5 concludes the work and discusses the result evaluation.

2. Related Works. The majority of classic SA investigations have used supervised machine learning techniques as their primary classification or clustering module [4]. These methods classified and displayed user-created texts that contained sentiment by utilizing n-gram features and bag-of-words (BOW) techniques [16]. These elements address the drawbacks of basic BOW strategies, such as failing to consider word order and syntactic structures [36]. Utilizing n-gram features is a disadvantage because the resulting feature space is dimensional, especially when n 3. Recently, feature selection strategies have been widely used to address this issue [35].

SVM, Naive Bayes (NB), and artificial neural networks (ANN) are a few of the often used techniques for extracting users' meanings from their texts and have provided promising results [19, 36, 24] . The supervised approaches have several drawbacks, including the fact that training can be time-consuming and slow at times. Numerous unsupervised lexicon-based approaches have been put forth to address these issues [15, 32]. These methods are quick, easy, and scalable. They are less efficient than their supervised counterparts, nevertheless, because they heavily rely on the lexicon [13, 1]. Lexicon-based techniques also suffer from field reliance, which limits their use in disciplines without specialized lexicons.

Aspect extraction [26] was addressed using a linear chain conditional random field after the author addressed aspect removal as a sequence classification issue. Text representations are separated from extracting features and model training in conventional methods (such as creating sentiment dictionaries), which concentrate on word embedding and extraction of features. Short texts include a high degree of randomness, confusion, inconsistency, and other qualities that make it easy for feature dispersal and contextual independence issues to arise during the processes of word embedding and extraction of features. When employing conventional sentiment analysis methods, all of these variables may result in decreased feature extraction accuracy and disconnection of contextual semantic links [34].

In SA, CNNs are used to extract specific characteristics. Those methods are helpful whenever the text is lengthy and particular local properties, such as n-grams, matter. For example, to analyze the sentiments at the data level, the researcher presented a CNN-based model that made use of optimal word embedding [12]. Their approach enhanced spatial, syntactic and semantic, and lexical features using pre-trained GloVe and Word2Vec embedding [25], but this work did not take into account the varying significance of terms and long dependency.

In the latest days, the attention mechanism was employed to improve DNN modeling optimization by letting the DNNs know where to focus their learning efforts. For example, [23] proposed a global pooling strategy architecture and one BiLSTM layer for binary sentiment categorization. The researcher [8] suggested a mixed model that incorporates Bi-LSTM, CNN, and the attention mechanism, AC-BiLSTM, for text categorization and information retrieval. On top of the word embedding layer, their models were using a one-dimensional CNN layer to automatically extract characteristics, a BiLSTM layer to extract long dependencies and an attention mechanism to concentrate on important text regions. The co-occurrence of both long and short dependency was not taken into account by the AC-BiLSTM framework.

Over the past several years, deep neural networks have become increasingly popular for solving pattern classification and machine learning challenges [14]. In the areas of speech recognition and computer vision, deep learning-based models have also demonstrated outstanding results [20, 17, 6]. Additionally, RNN-based networks have specifically had more success in the field of NLP deep networks [18, 22]. While conducting a sentiment analysis, word modeling is a crucial activity. Since the document level comprises more stored facts and is retrieved at the global level as opposed to the local level, a document-level sentiment classification is seen as a primary stage and is favored over sentence-level sentiment analysis. The amazing accomplishments in sentiment analysis that neural nets have achieved are due to their capacity to handle sequences of various lengths. Between these neural nets, the long short-term memory units are noteworthy [27, 36, 21, 24, 29]. Recently deep learning technologies with multi-layer neural networks are widely used to improve the performance of prediction and classification [10, 5].

3. Proposed Methodology. The proposed method for Aspect-based text classification uses the Attention Mechanism and RU-BiLSTM. Initially, the dataset is pre-processed and then the preprocessed data is given into feature extraction. Next, the extracted features are classified using RU-BiLSTM. We also suggested an attention mechanism-based LSTM and BiLSTM technique to categorize the polarity of reviews and tweets. The suggested approach aims to clarify the semantic similarities among features in addition to solving the issue of long-term dependency. The overall design of the suggested method is shown in Figure 3.1.

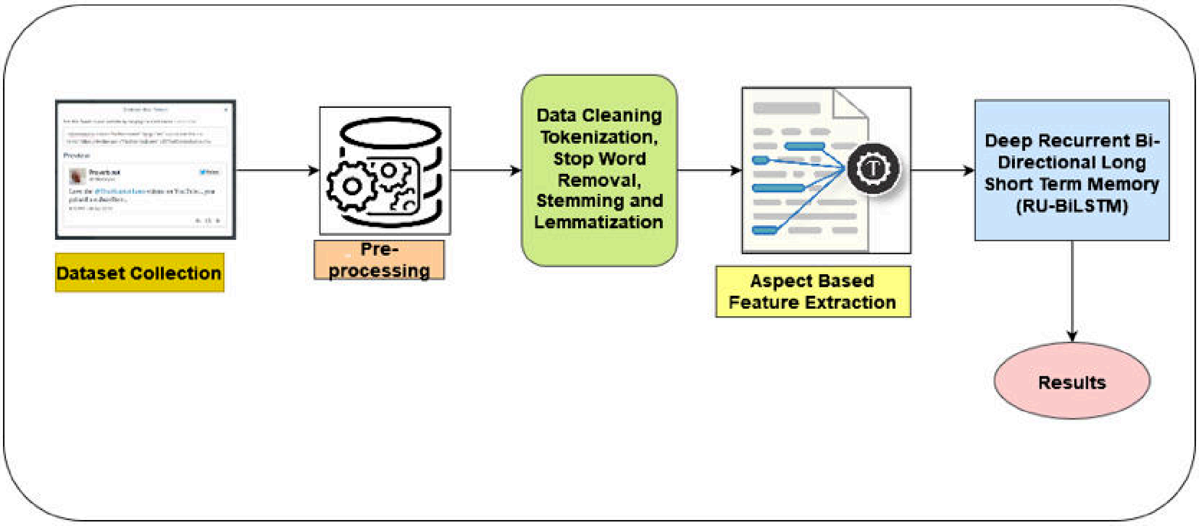

Fig. 3.1: Overall design of the Suggested Method

Table 3.1: Descriptions of Dataset [28]

Type of Dataset	Dataset Used	The total amount of Dataset	Positive	Negative
Reviews	App	752,748	133,897	112,774
	Kindle	995,619	59,548	58,157
	Electronics	1,730,148	199,865	198,912
	Cd	1,397,941	99,472	97,946
Tweets	Airline Tweets	15,649	3763	3483
	Sentiment 140	1,730,520	920,890	913,700

3.1. Dataset Collection. Long and short datasets were used in our study to perform sentiment analysis and aspect-based text categorization tasks. The following are the datasets' specifics:

1. APP: This collection for Android apps [10] includes 752,937 Amazon product reviews and related metadata.
2. Kindle: In this [10] the Kindle data contains 996,732 Amazon product reviews and related metadata. [10]
3. Electronics: 1,732,458 Amazon manufacturing remarks and related meta-data are included in this dataset for electronics [10].
4. CDs: The 1,097,592 products metadata and customer reviews for CDs and Vinyl [10] are from Amazon.
5. Airline Twitter: Here the dataset for airlines [10] is given and 14,641 tweets regarding significant issues with U.S. airlines are included in this dataset from February 2015.
6. Sentiment140: It consists of 1,730,520 tweets divided into positive and negative classes, and was created at Stanford University [10].

The parameters of the datasets utilized in the suggested model are shown in Table 3.1 along with additional information.

3.2. Data Preprocessing. The acquired data is cleansed during this procedure so that it is the sole data required for text classification. Preprocessing includes several steps, including review cleaning, which is useful for removing superfluous words and converting upper case letters in the text to lower case letters for text categorization, and stop words removal, which is useful for eliminating conjunction phrases as well as other special characters, like emoticons, that are not utilized for text classification, stemming, which is useful for turning all words into basic words, and tokenization, which is useful for tokenizing text.

3.3. Aspect-Based as Feature Extraction using Bag-of-Words and TF-IDF.

3.3.1. TF-IDF. An accepted feature extraction technique is the TF-IDF method. The TF-IDF extraction algorithm is much more accurate when compared to other algorithms. As a result, this work implemented vectorization analysis for reviews and tweets using the TF-IDF feature extraction technique. We are capable of determining whether another word was crucial in these textual samples by using the TF-IDF computation. The precise formula for calculation is as follows.

$$f(\omega) = TF(\omega) * IDF(\omega) = TF(\omega) * \frac{\log N}{n(\omega)+1} \tag{3.1}$$

After calculating the $TF(\omega)$ and $IDF(\omega)$ values independently, the overall weight value of TF-IDF is produced and arranged in ascending order. The dimensionality reduction process involved selecting the first five phrases with the greatest values. In Figure 3.2, the procedure is displayed.

3.3.2. TF-IDF Keyword Table. Numerous methods for numerical operations are available in the Python Scikit-learn machine learning toolkit, which also offers the Tfidf Transformer function needed for the TF-IDF method mentioned in this study. The weight was determined using the aforementioned method to eliminate keywords appropriate keywords 90 keywords from assessments of reviews and tweets were filtered out for this article, and then their corresponding values were determined.

3.3.3. Bag-of-Words. A method called bag of words is being used in natural language processing to count how many times each word appears in a text or review. Any arbitrary number of words, or n-grams, can be used to describe a phrase or token. The (1, 2) n-gram frequency is used in this investigation. The framing of unigrams, diagrams, and trigrams from a phrase is shown in Fig. 3.3. Due to the Bow model's consideration of all terms without taking into account the fact that some phrases are extremely consecutive in the corpus, a big matrix that is operationally costly to train is created.

3.4. Word Embeddings. It uses two types of word embeddings such as GloVe and Word2Vec. Here the sentence and aspects are embedded by the above methods.

3.4.1. GloVE . The input review or comment matrix $WE_g \in R^{n \times e}$ was made using a pre-trained GloVe embedding matrix, where e and n stand for the embedding dimensions and overall word count, accordingly. The max number of characters wt or the padding duration, $t \in [1, n]$, found in the remark for embedding purposes is repress ended $c \in R^n$, as seen below:

$$we_t = WE_g we_t, t \in [1, n] \tag{3.2}$$

3.4.2. Word2Vec. Setting low - dimensional feature-dense matrices for subsequent layers of neural networks is the main goal of embeddings. Our suggested method, RU-BiLSTM, can undertake detailed feature extraction with the use of n-grams and TF-IDF, improving classification even more. For improved text representation, a large volume of Chinese Comments text adjusted to a million terms was dragged from web reviews.

The input-padded segments are incorporated into the word embedding hidden layer before the data is fed to our main model. We offered a document D of M words, where $D = \{\mathrm{x}_1, \mathrm{x}_2, \ldots \mathrm{x}_M\}$, and x_i stands for each phrase in D. For each word in D, we used the embedded matrix W^k as a dictionary and lookup table, where $Wk \in R^{d^w|V|}$; in this case, V is our language dimension, which is set to 10,000, and d^w is our embedding size. While the user must supply the hyperparameter d^w, the weight vector Wk must be trained. This weight matrix's initialization is random. Word embedding's goal is to transform x_i into p_i, which rs essentially the result of the weight matrices and vectors.

$$p_i = W^k \cdot v^i \tag{3.3}$$

Here v^i is the vector of vocabulary size.

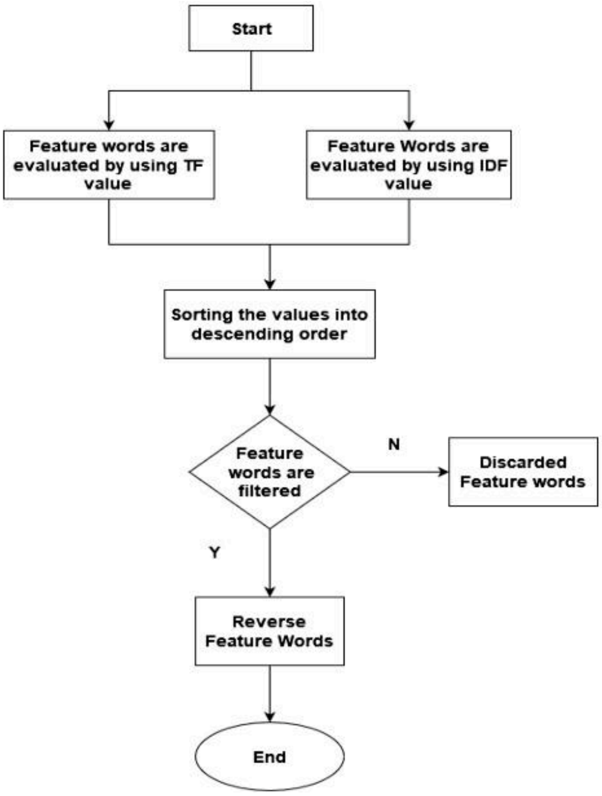

Fig. 3.2: Workflow of TF-IDF

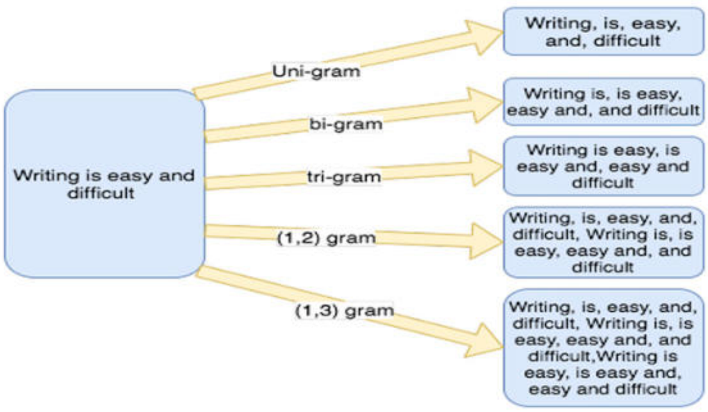

Fig. 3.3: Different types of grams used in texts

3.5. RU-BiLSTM. A more in-depth look at our model is also shown in Figure 3.4, where it is clear that BiLSTM encoding is first used at the aspect level before being used at the sentence level. Furthermore, both aspect and sentence levels employ an attentiveness strategy. The layers of our model RU-BiLSTM will be discussed below.

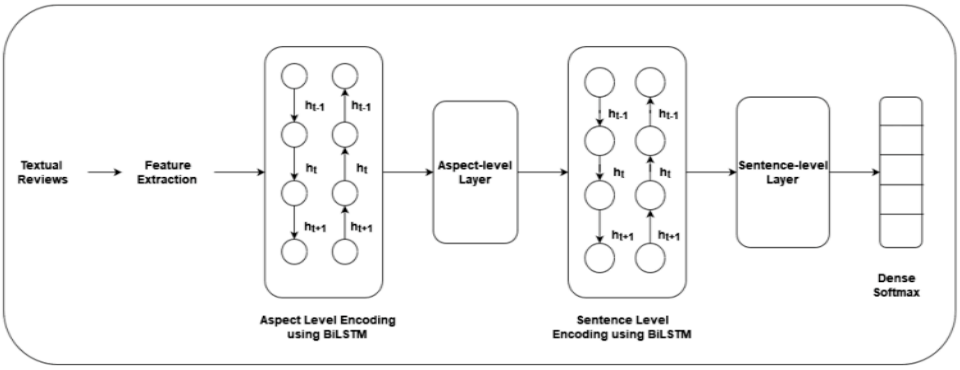

Fig. 3.4: Structure of Bi-directional Long Short Term Memory

3.5.1. Recurrent Neural Network. Both the traditional LSTM and its variation BiLSTM were empirically assessed. RNN can record historical data. In other words, while basic feed-forward networks just ahead the incoming input and lack memory, RNNs can maintain the prior outputs. Simple RNNs cannot recall long time stamps; they can only recall the details of small time stamps. Long-term dependency is the issue, and LSTM can be used to solve it. Modern LSTM devices can address this issue of long-term dependence. In addition to preserving context data, the recurrent design of LSTM can also solve the gradient descent's "blowing up and disappearing" problem.

The LSTM's gated components provide it the ability to manage the flow and select what to update and what to neglect. A recurrent module of single layer with a tanh squelching function makes up the simple RNN. The basic RNN can be theoretically described as follows as long as, input neurons as x_t, a hidden state in output is ht, and the preceding hidden state h_{t-1} as output.

$$h^t = \mathrm{g}_h(W_i x^t + W_R h^{(t-1)} + b_h) \tag{3.4}$$

$$y^t = \mathrm{g}_y(W_y h^t + b_y) \tag{3.5}$$

Here, weighted matrix is W, bias as b, hidden output as h^t, squashing function as GH, and it is the outcome.

Cell modes that function as conveyors are the main concept underpinning LSTM. This data is transmitted along the belt conveyor in a series of cycles, passing across various cell states with a minimal amount of linear interactions. Using gates, some data is added to or subtracted from the cell states throughout this journey. The traditional LSTM comprises four system memories, each with three multiplicative units called gates and a cell state c. As seen, the gates are input gate $i(t)$, output gate $o(t)$, and forget gate f_t.

$$f^t = \sigma(W_{xf} x_t + W_{hf} h^{t-1} + b_f) \tag{3.6}$$

$$i^t = \sigma(W_{xi} x_t + W_{hi} h^{t-1} + b_i) \tag{3.7}$$

$$c_t^{\sim} = \tanh(W_{xc} x_t + W_{hc} h^{t-1} + b_c) \tag{3.8}$$

The cell state being updated

$$c_t = f_t \bigotimes c_{t-1} + i_t \bigotimes c_t^{\sim} \tag{3.9}$$

$$o_t = \sigma(W_{xo} x_t + W_{hc} h^{t-1} + b_0) \tag{3.10}$$

$$h_t = o_t \bigotimes \tanh(c_t) \tag{3.11}$$

Thus, the sigmoid function is denoted by the symbol, while the forget gate, input gate, cell state, output gate, and hidden state are denoted by the symbols f_t, i_t, c_t, o_t, and h_t, correspondingly. In comparison, W indicates that b is the bias and that the weight matrix relates to every layer. Finding information that can be removed from the cell state and is not required is the first stage in LSTM. This decision is made by the sigmoid layer, also referred to as the forgotten layer.

The input gate layer, a sigmoid layer that determines the numbers to be modified, and the layer, that creates an angle of potential possible numbers to be placed in cell state $c \bigotimes t$, are two-pass layers that we use to decide whether additional knowledge is to be supplied. As noted in Equation 3.9, the new state of cell ct is then merged with the previous state of cell ct_1. The final step is to run a sigmoid layer between 0 and 1 to decide which parts are included in the cell state as the output of the algorithm, which is then multiplied based on element-wise (represented by the symbol) with a layer tanh that takes the cell state as a dispute. The function 'tanh' limits the range of numbers to those between 0 and 1. As a result, concealed output h_t.

3.5.2. Attention mechanism with Bidirectional Network. One shortcoming of the traditional LSTM is that it only takes into account past outputs. We may think about the environment in both ways and utilize the past and the future contexts thanks to our Bi-LSTM driven by the attention mechanism. This is accomplished by using two distinct hidden bidirectional layers that are later merged into a single layer output and sent into the attention layer. The forward and backward layers of our bidirectional network, which are designated as L_f and L_b, accordingly, are hidden LSTM layers. The two layers move back and forth through the SC feature sequences from 1 to n. Equations 3.12 and 3.13 provide a mathematical representation of the network.

$$\overrightarrow{h}_f = \overrightarrow{L}_f(SC)_n, \quad n \in [1, 100] \tag{3.12}$$

$$\overleftarrow{h}_b = \overleftarrow{L}_b(SC)_n, \quad n \in [1, 100] \tag{3.13}$$

The BiLSTM layer decides how words are annotated in both directions before summarizing the data. The attention mechanism enables one to focus on a phrase that is more significant to a particular class of sentiment content. Key features are given more weight by the attention layer, whereas unimportant features are ignored. The thick softmax layer continues to process the gathered characteristics before producing the final result.

Attention Mechanism. Every output h_t is from hidden layer of LSTM layers is subjected to the attention mechanism to give words with different contributions distinct weights. One common method of giving various weights to several words in a sentence is to use a weighted mixture of all hidden states. The following is how the context vector is to be calculated using the attention model:

$$\mu_t = \tanh(W.h_t + b) \tag{3.14}$$

$$\alpha_t = \frac{\exp(\mu_t^T u_w)}{\sum_t \exp(\mu_t^T u_w)} \tag{3.15}$$

$$c = \sum_t^T \alpha_t h_t \tag{3.16}$$

Here u_w represents the vector representation which is initialized arbitrarily and eventually learned at training stage, and u_t represents a concealed version of h_t. The input signal has a total of T time steps, and each time step t has a weight calculated for every state h_t. After that, the weightings α_t are calculated as given in Equation 3.14. These important weights are gradually combined into c as stated in Equation by using a weight value to them 3.16. c symbolizes the vector and combines all of the textual review's word data.

Table 4.1: Simulation Parameters

Parameters Used	**Values**
Dimensions (D)	100
Hidden Bi-LSTM units	155
Total comments for reviews	100
Total comments for tweets	55
Activation Function	Relu
Optimizer	Softmax
Kernel-size	4
Convolution Layer	64
Loss function	Binary-cross entropy

3.5.3. Softmax layer. The vector V_{gap} output was quickly passed into the Softmax layer to perform sentiment analysis, as illustrated in the formula below:

$$\hat{y} = softmax(WV_{gap} + b) \tag{3.17}$$

The goal of cross-entropy was initiated to represent the discrepancy between the projected emotional category y and the actual sentimental category y to assess the suggested model.

$$Loss = -\sum_i y_i log\widehat{y_i}$$

Here i stand for the phrase's index.

4. Result Analysis. This section shows the tests that were carried out to evaluate the RU-BiLSTM model's SA and aspect-based text classification results on several testing samples. Following a discussion of the findings is a description of the experimental design and foundational procedures.

4.1. Simulation Parameters. Tensorflow 1.13.1 with Keras 2.24 libraries built in Python 3.7.1 and an Ubuntu 16.04 machine with a Core Tetranuclear i7-7700k CPU and a GTX1080 Ti GAMING X 11GB GPU were used to apply the RU-BiLSTM model. The Tokenizer approach employs 100,000 words to build the input comment matrix C. We fixed the padding values to 45 and 100, accordingly, assuming that the first 45 and 100 words of remarks in the tweet and review datasets, respectively. The pre-trained, publicly accessible GloVe and word2Vec models were used in the current investigation as the embedding layer's weights. It used the "Gigaword 5 + Wikipedia 2014" version of GloVe, which has a vocabulary size of 400,000 words and six billion tokens. The embedding length of 300 was employed for the embedding layer. Table 4.1 displays further parameter selections made use of in the suggested model.

For assessing the effectiveness of the suggested approach, four assessment principles—Accuracy, Precision, Recall, and F1 measures—were used. In tasks involving SA and aspect-based text classification, these standards are frequently used. This is how these standards are calculated:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} X100 \tag{4.1}$$

$$precision = \frac{TP}{TP + FP} X100 \tag{4.2}$$

$$recall = \frac{TP}{TP + FN} \tag{4.3}$$

Table 4.2: Experimental outcome of Kindle Dataset

Techniques Used	Classes Used	Accuracy	Precision	Recall	F1-Score
SS-BED	Positive	0.8910	0.9461	0.8308	0.8827
	Negative		0.8521	0.9514	0.8979
AC-Bi-LSTM	Positive	0.9074	0.9018	0.8553	0.9018
	Negative		0.9122	0.9595	0.9122
DNN-WHAT	Positive	0.9372	0.9612	0.9123	0.9377
	Negative		0.9193	0.9614	0.9343
RU-BiLSTM	**Positive**	**0.9585**	**0.9732**	**0.9256**	**0.9478**
	Negative		**0.9317**	**0.9775**	**0.9495**

Table 4.3: Experimental results of APP dataset

Techniques Used	Classes Used	Accuracy	Precision	Recall	F1-Score
SS-BED	Positive	0.8370	0.9371	0.8778	0.8127
	Negative		0.8731	0.9014	0.8639
AC-Bi-LSTM	Positive	0.9104	0.9428	0.9012	0.9074
	Negative		0.9512	0.9075	0.9174
DNN-MHAT	Positive	0.9256	0.9505	0.9006	0.9244
	Negative		0.9063	0.9523	0.9304
Proposed RU-BiLSTM	**Positive**	**0.9397**	**0.9624**	**0.9106**	**0.9318**
	Negative		**0.9210**	**0.9605**	**0.9402**

$$F1-score = \frac{2*Precision*Recall}{Precision+Recall} \tag{4.4}$$

The proposed RU-BiLSTM method is compared with existing methods such as DNN-MHAT, AC-BiLSTM and SS-BED. It uses long and short datasets for evaluation.

4.2. Evaluation results of Long Reviews. In this long review, four types of datasets were used such as Kindle, APP, CD, and Electronics datasets. The positive and negative aspects were classified. Tables 4.2, 4.3, 4.4, 4.5 show the experimental results of the review dataset.

The table 4.2 shows the results of evaluating four different techniques (SS-BED, AC-Bi-LSTM, DNN-WHAT, and Proposed RU-BiLSTM) on a binary classification task, where the classes are Positive and Negative. The evaluation metrics used are Accuracy, Precision, Recall, and F1-Score for both classes. The results show that all techniques performed well, with the Proposed RU-BiLSTM achieving the highest overall performance with an Accuracy of 0.9585. It also achieved the highest scores for Precision, Recall, and F1-Score for the Positive class, indicating that it was able to correctly identify the Positive class with a high level of precision and recall.

The table 4.3 presents experimental results of four different techniques used to classify classes of the APP dataset. The dataset consists of text data and the techniques used are SS-BED, AC-Bi-LSTM, DNN-MHAT, and Proposed RU-BiLSTM. The SS-BED technique achieved an accuracy of 0.8370 for the positive class and 0.8731 for the negative class. The precision for the positive class was 0.9371, and the recall was 0.8778, resulting in an F1-score of 0.8127. For the negative class, the precision was 0.9014, and the recall was 0.8639, resulting in an F1-score of 0.8370. The AC-Bi-LSTM technique achieved an accuracy of 0.9104 for the positive class and 0.9512 for the negative class. The precision for the positive class was 0.9428, and the recall was 0.9012, resulting in an F1-score of 0.9074. For the negative class, the precision was 0.9075, and the recall was 0.9174, resulting in an F1-score of 0.9124.

The DNN-MHAT technique achieved an accuracy of 0.9256 for the positive class and 0.9063 for the negative class. The precision for the positive class was 0.9505, and the recall was 0.9006, resulting in an F1-score of

Table 4.4: Experimental results of CD Dataset

Techniques Used	Classes Used	Accuracy	Precision	Recall	F1-Score
SS-BED	Positive	0.8512	0.8971	0.8568	0.8723
	Negative		0.8798	0.8914	0.8930
AC-Bi-LSTM	Positive	0.8704	0.9208	0.8712	0.8474
	Negative		0.9012	0.8975	0.8974
DNN-MHAT	Positive	0.8913	0.9188	0.8578	0.8706
	Negative		0.8692	0.9306	0.8959
Proposed RU-BiLSTM	**Positive**	**0.9071**	**0.9243**	**0.8706**	**0.9089**
	Negative		**0.8712**	**0.9434**	**0.9207**

Table 4.5: Experimental Results of Electronic Dataset

Techniques Used	Classes Used	Accuracy	Precision	Recall	F1-Score
SS-BED	Positive	0.8512	0.8671	0.8668	0.8833
	Negative		0.8818	0.8812	0.8607
AC-Bi-LSTM	Positive	0.8704	0.9078	0.8904	0.8704
	Negative		0.8972	0.8741	0.8520
DNN-MHAT	Positive	0.9112	0.9411	0.8777	0.9065
	Negative		0.8821	0.9482	0.9127
Proposed RU-BiLSTM	**Positive**	**0.9245**	**0.9537**	**0.8837**	**0.9189**
	Negative		**0.8977**	**0.9599**	**0.9227**

0.9244. For the negative class, the precision was 0.9523, and the recall was 0.9304, resulting in an F1-score of 0.9412. The Proposed RU-BiLSTM technique achieved the highest accuracy of 0.9397 for the positive class and 0.9210 for the negative class. The precision for the positive class was 0.9624, and the recall was 0.9106, resulting in an F1-score of 0.9318. For the negative class, the precision was 0.9605, and the recall was 0.9402, resulting in an F1-score of 0.9502.

The table 4.4 presents the experimental results of four different techniques used to classify classes of the CD dataset. The techniques used are SS-BED, AC-Bi-LSTM, DNN-MHAT, and Proposed RU-BiLSTM. The SS-BED technique achieved an accuracy of 0.8512 for the positive class and 0.8798 for the negative class. The precision for the positive class was 0.8971, and the recall was 0.8568, resulting in an F1-score of 0.8723. For the negative class, the precision was 0.8914, and the recall was 0.8930, resulting in an F1-score of 0.8922.

The AC-Bi-LSTM technique achieved an accuracy of 0.8704 for the positive class and 0.9012 for the negative class. The precision for the positive class was 0.9208, and the recall was 0.8712, resulting in an F1-score of 0.8474. For the negative class, the precision was 0.8975, and the recall was 0.8974, resulting in an F1-score of 0.8975. The DNN-MHAT technique achieved an accuracy of 0.8913 for the positive class and 0.8692 for the negative class. The precision for the positive class was 0.9188, and the recall was 0.8578, resulting in an F1-score of 0.8706. For the negative class, the precision was 0.9306, and the recall was 0.8959, resulting in an F1-score of 0.9130.

The Proposed RU-BiLSTM technique achieved the highest accuracy of 0.9071 for the positive class and 0.8712 for the negative class. The precision for the positive class was 0.9243, and the recall was 0.8706, resulting in an F1-score of 0.9089. For the negative class, the precision was 0.9434, and the recall was 0.9207, resulting in an F1-score of 0.9320.

The table 4.5 shows the experimental results of different techniques used on an electronic dataset, where each technique is evaluated based on its accuracy, precision, recall, and F1-score for positive and negative classes. The first technique used is SS-BED, which achieved an accuracy of 0.8512 for the positive class and 0.8818 for the negative class. Its precision is 0.8671 for the positive class and 0.8812 for the negative class,

Fig. 4.1: Results of Kindle dataset

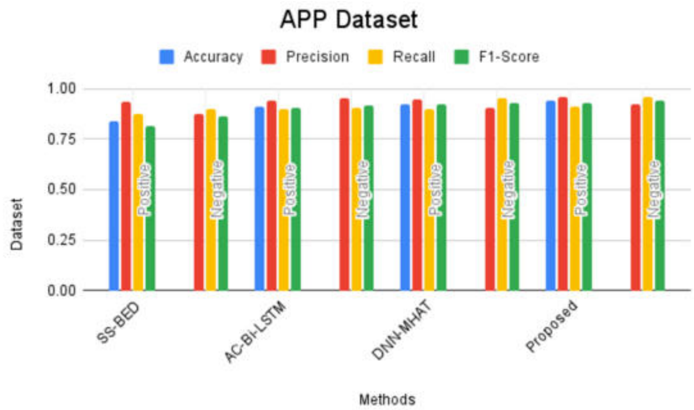

Fig. 4.2: Results of APP dataset

while its recall is 0.8668 for the positive class and 0.8607 for the negative class. Its F1-score is 0.8833 for the positive class and 0.8520 for the negative class.

The second technique is AC-Bi-LSTM, which achieved an accuracy of 0.8704 for the positive class and 0.8972 for the negative class. Its precision is 0.9078 for the positive class and 0.8741 for the negative class, while its recall is 0.8904 for the positive class and 0.8520 for the negative class. Its F1-score is 0.8704 for the positive class and 0.8520 for the negative class. The third technique is DNN-MHAT, which achieved an accuracy of 0.9112 for the positive class and 0.8821 for the negative class. Its precision is 0.9411 for the positive class and 0.9482 for the negative class, while its recall is 0.8777 for the positive class and 0.9127 for the negative class. Its F1-score is 0.9065 for the positive class and 0.9127 for the negative class.

Finally, the proposed technique RU-BiLSTM achieved the highest accuracy of 0.9245 for positive class and 0.8977 for negative class. Its precision is 0.9537 for positive class and 0.9599 for negative class, while its recall is 0.8837 for positive class and 0.9227 for negative class. Its F1-score is 0.9189 for positive class and 0.9227 for the negative class.

Figures 4.1, 4.2, 4.3 and 4.4 show the experimental results of Long dataset such as Kindle, App, CD and Electronic datasets. It is compared with existing algorithms. Among all these methods, the suggested method achieves greater performance in terms of accuracy, precision, recall, and f1-score.

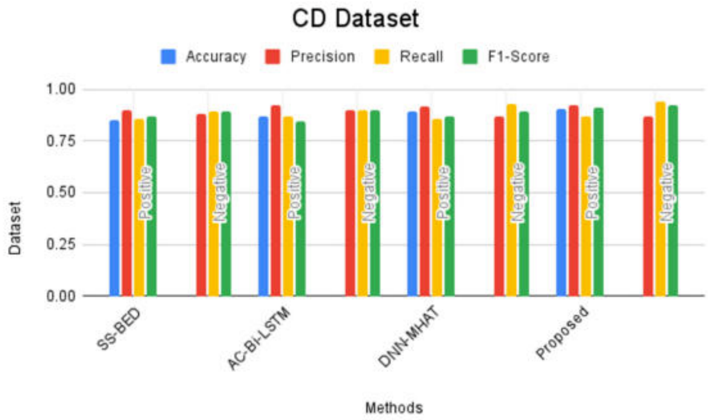

Fig. 4.3: Results of CD dataset

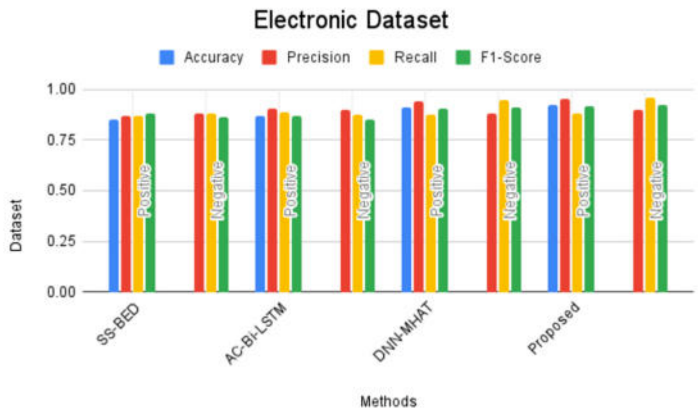

Fig. 4.4: Results of Electronic Dataset

4.3. Evaluation of Short Tweets. In short tweets, two types of datasets were used as Airline Twitter dataset and the Sentiment 140 dataset. The positive and negative aspects were classified. Table 4.6 and 4.7 shows the experimental outcome of two tweet datasets.

The table 4.7 displays the experimental results of four different methods applied to the Sentiment 140 dataset, which is a collection of tweets labeled as positive or negative. The methods used are SS-BED, AC-Bi-LSTM, DNN-MHAT, and Proposed RU-BiLSTM, and the evaluation metrics used are Accuracy, Precision, Recall, and F1-Score for both positive and negative classes.

The results show that the Proposed RU-BiLSTM method outperforms the other methods in terms of accuracy, precision, recall, and F1-Score for both positive and negative classes. It achieved an accuracy of 0.8732 for positive and 0.8967 for negative classes, which are the highest among all the methods. The precision, recall, and F1-Score for positive and negative classes are also high, indicating that the Proposed RU-BiLSTM method is effective in classifying tweets as positive or negative.

In Figures 4.5 and 4.6 demonstrate how well RU-NiLSTM performed in terms of accuracy on the Sentiment140 and Airline Twitter datasets, with accuracy scores of 0.39 and 0.45, correspondingly. Upon this Sentiment140 and Airline Twitter datasets, the enhancements for the F1 scale are 0.72% and 0.63% for the positive classes and 0.37% and 0.53% for the negative classes, correspondingly. As shown before, our RU-BiLSTM performed much better than the other techniques in terms of accuracy and F1 scale.

Table 4.6: Experimental outcomes of Airline Twitter Dataset

Techniques Used	Classes Used	Accuracy	Precision	Recall	F1-Score
SS-BED	Positive	0.8789	0.8211	0.8748	0.8673
	Negative		0.8518	0.8312	0.8307
AC-Bi-LSTM	Positive	0.8914	0.8778	0.8714	0.8504
	Negative		0.9072	0.8741	0.8120
DNN-MHAT	Positive	0.9302	0.9588	0.9608	0.9603
	Negative		0.8411	0.8167	0.8261
Proposed RU-BiLSTM	**Positive**	**0.9475**	**0.9598**	**0.9789**	**0.9752**
	Negative		**0.8567**	**0.8321**	**0.8467**

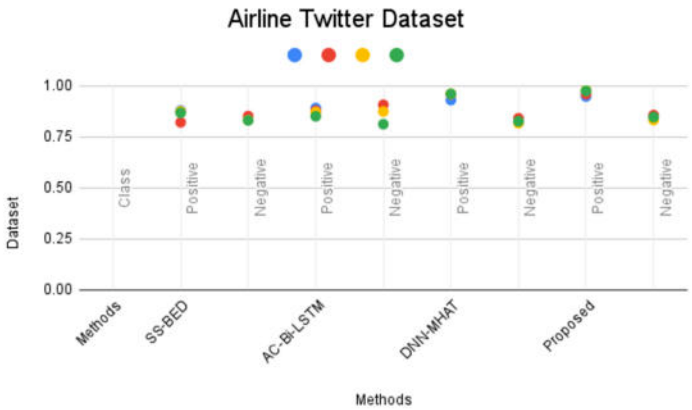

Fig. 4.5: Results of Airline Twitter Dataset

5. Conclusion. In this study, we present a deep recurrent Bi-directional Long-short Term Memory model for an aspect-level-based attention mechanism for text sentiment analysis. Initially, the data is collected and then preprocessed using Tokenization, Stop word removal, Stemming, and Lemmatization. After pre-processing, the features are vectorized by using Bag-of-Words and TF-IDF. Then, the extracted features are given into word embeddings by GloVe and Word2vec. It uses Deep Recurrent based Bidirectional Long Short Term Memory (RU-BiLSTM) for aspect-based sentiment analysis. The RU-Bi-LSTM method integrates aspect-based embeddings and an attention mechanism for text classification. Finally, the binary and ternary classification outcomes are obtained using the final dense softmax output layer. In this proposed method, four long review dataset and two short Twitter dataset is used. It classifies the positive and negative aspects based texts. Lastly, it was discovered that the RU-BiLSTM model developed in this work greatly outperformed and compared with DNN-MHAT, AC-BiLSTM, and SS-BED. The proposed RU-BiLSTM outperforms better in terms of accuracy, precision, recall, and f1-score. The proposed model is trained and evaluated only on English text datasets, and it may not be directly applicable to other languages. Further, the model only focuses on aspect-level sentiment analysis and may not be suitable for document-level sentiment analysis. In the future, The proposed model can be extended to support multilingual text classification to improve its applicability to other languages. Also, the proposed model can be extended to incorporate other types of attention mechanisms such as self-attention and transformer-based attention.

REFERENCES

[1] M. Al-Ayyoub, A. A. Khamaiseh, Y. Jararweh, and M. N. Al-Kabi, *A comprehensive survey of arabic sentiment analysis*,

Table 4.7: Experimental results of Sentiment 140 Dataset

Methods Used	Class	Accuracy	Precision	Recall	F1-Score
SS-BED	Positive	0.8789	0.8311	0.8048	0.7773
	Negative		0.7518	0.8312	0.8307
AC-Bi-LSTM	Positive	0.7814	0.7478	0.8714	0.7904
	Negative		0.8172	0.8741	0.8120
DNN-MHAT	Positive	0.8217	0.7782	0.7585	0.8092
	Negative		0.8896	0.7478	0.8212
Proposed RU-BiLSTM	**Positive**	**0.8732**	**0.8098**	**0.9137**	**0.8542**
	Negative		**0.8967**	**0.7721**	**0.8367**

Fig. 4.6: Results of Sentiment140

Information processing & management, 56 (2019), pp. 320–342.

[2] M. Alam, F. Abid, C. Guangpei, and L. Yunrong, *Social media sentiment analysis through parallel dilated convolutional neural network for smart city applications*, Computer Communications, 154 (2020), pp. 129–137.

[3] A. H. Alamoodi, B. Zaidan, M. Al-Masawa, S. M. Taresh, S. Noman, I. Y. Ahmaro, S. Garfan, J. Chen, M. A. Ahmed, A. Zaidan, et al., *Multi-perspectives systematic review on the applications of sentiment analysis for vaccine hesitancy*, Computers in Biology and Medicine, 139 (2021), p. 104957.

[4] N. Ayoobi, D. Sharifrazi, R. Alizadehsani, A. Shoeibi, J. M. Gorriz, H. Moosaei, A. Khosravi, S. Nahavandi, A. G. Chofreh, F. A. Goni, et al., *Time series forecasting of new cases and new deaths rate for covid-19 using deep learning methods*, Results in physics, 27 (2021), p. 104495.

[5] P. Bala Anki Reddy, S. Jakeer, H. Thameem Basha, S. R. Reddisekhar Reddy, and T. Mahesh Kumar, *Multi-layer artificial neural network modeling of entropy generation on mhd stagnation point flow of cross-nanofluid*, Waves in Random and Complex Media, (2022), pp. 1–28.

[6] M. Birjali, M. Kasri, and A. Beni-Hssane, *A comprehensive survey on sentiment analysis: Approaches, challenges and trends*, Knowledge-Based Systems, 226 (2021), p. 107134.

[7] B. A. Chandio, A. S. Imran, M. Bakhtyar, S. M. Daudpota, and J. Baber, *Attention-based ru-bilstm sentiment analysis model for roman urdu*, Applied Sciences, 12 (2022), p. 3641.

[8] A. Chatterjee, U. Gupta, M. K. Chinnakotla, R. Srikanth, M. Galley, and P. Agrawal, *Understanding emotions in text using deep learning and big data*, Computers in Human Behavior, 93 (2019), pp. 309–317.

[9] Y. Cheng, L. Yao, G. Zhang, T. Tang, G. Xiang, H. Chen, Y. Feng, and Z. Cai, *Text sentiment orientation analysis of multi-channels cnn and bigru based on attention mechanism*, J. Comput. Res. Dev, 57 (2020), pp. 2583–2595.

[10] A. de Haro-García, G. Cerruela-García, and N. García-Pedrajas, *Ensembles of feature selectors for dealing with class-imbalanced datasets: A proposal and comparative study*, Information Sciences, 540 (2020), pp. 89–116.

[11] Z. Gao, Z. Li, J. Luo, and X. Li, *Short text aspect-based sentiment analysis based on cnn+ bigru*, Applied Sciences, 12 (2022), p. 2707.

[12] K. Garcia and L. Berton, *Topic detection and sentiment analysis in twitter content related to covid-19 from brazil and the usa*, Applied soft computing, 101 (2021), p. 107057.

[13] A. Ghafoor, A. S. Imran, S. M. Daudpota, Z. Kastrati, R. Batra, M. A. Wani, et al., *The impact of translating resource-rich datasets to low-resource languages through multi-lingual text processing*, IEEE Access, 9 (2021), pp. 124478–124490.
[14] Z. Hameed and B. Garcia-Zapirain, *Sentiment classification using a single-layered bilstm model*, Ieee Access, 8 (2020), pp. 73992–74001.
[15] F. Hemmatian and M. K. Sohrabi, *A survey on classification techniques for opinion mining and sentiment analysis*, Artificial intelligence review, 52 (2019), pp. 1495–1545.
[16] T. Hou and L. Zhou, *Chinese ship fault relation extraction method based on bidirectional gru neural network and attention mechanism*, Comput. Sci, 48 (2021), pp. 154–158.
[17] A. S. Imran, S. M. Daudpota, Z. Kastrati, and R. Batra, *Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on covid-19 related tweets*, Ieee Access, 8 (2020), pp. 181074–181090.
[18] Z. Kastrati, F. Dalipi, A. S. Imran, K. Pireva Nuci, and M. A. Wani, *Sentiment analysis of students' feedback with nlp and deep learning: A systematic mapping study*, Applied Sciences, 11 (2021), p. 3986.
[19] H. Kwon and S. Lee, *Textual backdoor attack for the text classification system*, Security and Communication Networks, 2021 (2021), pp. 1–11.
[20] A. Ligthart, C. Catal, and B. Tekinerdogan, *Systematic reviews in sentiment analysis: a tertiary study*, Artificial Intelligence Review, (2021), pp. 1–57.
[21] S. Maheswaran, S. Sathesh, P. Priyadharshini, and B. Vivek, *Identification of artificially ripened fruits using smart phones*, in 2017 international conference on intelligent computing and control (I2C2), IEEE, 2017, pp. 1–6.
[22] S. Maheswaran, S. Sathesh, G. Saran, and B. Vivek, *Automated coconut tree climber*, in 2017 International Conference on Intelligent Computing and Control (I2C2), IEEE, 2017, pp. 1–6.
[23] K. Mehmood, D. Essam, K. Shafi, and M. K. Malik, *Discriminative feature spamming technique for roman urdu sentiment analysis*, IEEE Access, 7 (2019), pp. 47991–48002.
[24] A.-S. Mohammad, M. M. Hammad, A. Sa'ad, A.-T. Saja, and E. Cambria, *Gated recurrent unit with multilingual universal sentence encoder for arabic aspect-based sentiment analysis*, Knowledge-Based Systems, (2021), p. 107540.
[25] A. B. Nassif, A. Elnagar, I. Shahin, and S. Henno, *Deep learning for arabic subjective sentiment analysis: Challenges and research opportunities*, Applied Soft Computing, 98 (2021), p. 106836.
[26] S. Rani and P. Kumar, *Deep learning based sentiment analysis using convolution neural network*, Arabian Journal for Science and Engineering, 44 (2019), pp. 3305–3314.
[27] Z. Safdar, R. S. Bajwa, S. Hussain, H. B. Abdullah, K. Safdar, and U. Draz, *The role of roman urdu in multilingual information retrieval: A regional study*, The Journal of Academic Librarianship, 46 (2020), p. 102258.
[28] H. S. Sharaf Al-deen, Z. Zeng, R. Al-sabri, and A. Hekmat, *An improved model for analyzing textual sentiment based on a deep neural network using multi-head attention mechanism*, Applied System Innovation, 4 (2021), p. 85.
[29] B. Vivek, S. Maheswaran, P. Keerthana, S. Sathesh, S. Bringeraj, R. A. Sri, and S. A. Sulthana, *Low-cost raspberry pi oscilloscope*, in 2018 international conference on intelligent computing and communication for smart World (I2C2SW), IEEE, 2018, pp. 386–390.
[30] L. Wang, C. Liu, D. CAI, and T. Lu, *Chinese text sentiment analysis based on cnn-bigru network with attention mechanism*, Journal of Computer Applications, 39 (2019), p. 2841.
[31] W. Wang, Y. Sun, Q. Qi, and X. Meng, *Text sentiment classification model based on bigru-attention neural network*, Appl. Res. Comput, 36 (2019), pp. 3558–3564.
[32] A. Yadav and D. K. Vishwakarma, *Sentiment analysis using deep learning architectures: a review*, Artificial Intelligence Review, 53 (2020), pp. 4335–4385.
[33] S. Yao, P. Jing, P. Li, and J. Chen, *A multi-dimension traceable privacy-preserving prevention and control scheme of the covid-19 epidemic based on blockchain*, Connection Science, 34 (2022), pp. 1654–1677.
[34] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, *A survey of sentiment analysis in social media*, Knowledge and Information Systems, 60 (2019), pp. 617–663.
[35] B. Zhang, X. Zhou, Y. Luo, H. Zhang, H. Yang, J. Ma, and L. Ma, *Opportunities and challenges: Classification of skin disease based on deep learning*, Chinese Journal of Mechanical Engineering, 34 (2021), pp. 1–14.
[36] X. Zhang, M. Zhang, X. Li, et al., *Rolling bearing fault mode recognition based on 2d image and cnn-bigru*, J. Vib. Shock, 40 (2021), pp. 194–201.

Edited by: Sathishkumar V E
Special Issue on: Scalability and Sustainability in Distributed Sensor Networks
Received: Feb 26, 2023
Accepted: May 2, 2023