applied sciences

MDPI

Article

Sentiment Analysis of Text Reviews Using Lexicon-Enhanced Bert Embedding (LeBERT) Model with Convolutional Neural Network

James Mutinda [1,*], Waweru Mwangi [2] and George Okeyo [3]

[1] Department of Research and Consultancy, Kenya School of Government-Embu Campus, Embu P.O. Box 402-60100, Kenya
[2] Department of Computing, Jomo Kenyatta University of Agriculture & Technology, Nairobi P.O. Box 62000-00200, Kenya
[3] College of Engineering, Carnegie Mellon University Africa, Kigali Innovation City, Bumbogo, Kigali BP 6150, Rwanda
* Correspondence: james.mutinda@ksg.ac.ke

Abstract: Sentiment analysis has become an important area of research in natural language processing. This technique has a wide range of applications, such as comprehending user preferences in ecommerce feedback portals, politics, and in governance. However, accurate sentiment analysis requires robust text representation techniques that can convert words into precise vectors that represent the input text. There are two categories of text representation techniques: lexicon-based techniques and machine learning-based techniques. From research, both techniques have limitations. For instance, pre-trained word embeddings, such as Word2Vec, Glove, and bidirectional encoder representations from transformers (BERT), generate vectors by considering word distances, similarities, and occurrences ignoring other aspects such as word sentiment orientation. Aiming at such limitations, this paper presents a sentiment classification model (named LeBERT) combining sentiment lexicon, N-grams, BERT, and CNN. In the model, sentiment lexicon, N-grams, and BERT are used to vectorize words selected from a section of the input text. CNN is used as the deep neural network classifier for feature mapping and giving the output sentiment class. The proposed model is evaluated on three public datasets, namely, Amazon products' reviews, Imbd movies' reviews, and Yelp restaurants' reviews datasets. Accuracy, precision, and F-measure are used as the model performance metrics. The experimental results indicate that the proposed LeBERT model outperforms the existing state-of-the-art models, with a F-measure score of 88.73% in binary sentiment classification.

Keywords: natural language processing; word embeddings; BERT; sentiment analysis; convolutional neural network; sentiment lexicon

Citation: Mutinda, J.; Mwangi, W.; Okeyo, G. Sentiment Analysis of Text Reviews Using Lexicon-Enhanced Bert Embedding (LeBERT) Model with Convolutional Neural Network. *Appl. Sci.* **2023**, *13*, 1445. https://doi.org/10.3390/app13031445

Academic Editors: Xiangjie Kong, Wei Wang and Han Liu

Received: 24 December 2022
Revised: 12 January 2023
Accepted: 13 January 2023
Published: 21 January 2023

1. Introduction

Recently, social media platforms have created opportunities for businesses and organizations to obtain feedback from their customers and clients through reviews in the form of user-generated posts. Such posts are availed through social media and worldwide web in form of blogs, which contain data in text, audio, visual, or a combination of the three modes. Specifically, social media text data are characterized by short sentences, which are unstructured, semi-structured, and normally full of colloquial language, making it messy, difficult, and time consuming to build its vector representations and sentiment classification [1–4]. However, through sentiment analysis (SA), one of the big data analytics techniques, the text data can provide insightful business information [4]. Sentiment analysis is the process of classifying texts into predetermined opinion classes [3], which can be performed at document level, sentence level, or word level. Sentence level SA is a text classification task that assigns short texts (sentences) to predefined sentiment or opinion

classes. Sentiment analysis of social media data is a sentence level SA task since most posts are very short usually less than forty (40) words. Currently, few tools can perform sentiment analysis of social media text data effectively [4]. This is attributed to the nature of social media texts, which are unstructured, making it difficult to extract the right features at the text representation phase. According to Zhiying Jiang et al. [1], text representation is the second phase in sentiment analysis after text data preprocessing. In this phase, documents or sentences are converted into numeric vectors that represent the texts by use of vector space models (VSM).

Conversion of text to vector representation is the cornerstone of text classification models [5]. The accuracy and efficiency of sentiment analysis is dependent on whether or not the word vector is representative of the text [5–7]. From the literature, there are two widely used text-vector representation techniques: (1) natural language processing (NLP) techniques based on bag of words, part of speech (POS) tags, and sentiment lexicons [1,8–10]; (2) deep learning-based automated vector representation approaches such as word embeddings [11–13]. Word Embedding is one of the most useful deep learning methods used for constructing vector representations of words and documents in text classification tasks. This is because of their abilities to capture the syntactic and semantic relations among words [14]. Word embeddings models are based on deep learning Word2Vec [15], global vectors (Glove) [16], FastText [17], and bidirectional encoder representations from transformers (BERT) model [18]. Although these word embeddings methods are very effective compared to conventional NLP-based methods [19,20], they have some limitations and thus need improvement. For instance, effective training and vector representation of words and word embeddings require a very large corpus. Due to these limitations, researchers use pre-trained word embeddings for transfer learning, which may not correspond well with their data, especially small-sized datasets [21]. Further, the pre-trained word embeddings vectors do not consider the context of the word or other characteristics of the word, such as semantic orientation of the word. Existing NLP techniques, such as sentiment lexicon, POS tags, and word positions, can be used to improve performance of sentiment analysis models based on word embeddings [14].

In this paper, we propose a deep learning-based sentiment analysis model for user reviews, which combines sentiment lexicon, N-grams, and BERT word embeddings. In the model, we combine pre-trained word embeddings with sentiment lexicon to generate word representation for sentiment analysis. A text (review, sentence, or a document) is treated as a collection of word N-grams, and a sentiment lexicon is used to identify a section (N-grams) of the text where a sentiment may be found. BERT pre-trained word embeddings are then used to build vector representation of the text. In addition to solving the aforementioned limitations of word embeddings, our model reduces high feature dimensionality and computational costs brought by building word vectors from the entire text. We evaluate the proposed approach on the Yelp datasets in which the experimental results show that the model improves accuracy of pre-trained word embeddings. The main contribution of this paper, therefore, is to advance utilization of BERT pre-trained word embeddings model for sentiment analysis. We noted that BERT is one of the state-of-the-art models for building word vectors for NLP tasks, such as sentiment analysis. The novelty of the proposed model is the use of sentiment lexicon with N-grams to identify a section of input text, such as a review where sentiment is likely to be found. This approach proactively reduces feature dimensions of word vectors in the embedding layer of deep learning models such as CNN.

The rest of the paper is organized as follows. Section 2 presents related work. The proposed approach is described in Section 3. Section 4 describes the experimental procedures carried out. The result and discussion are presented in Section 5. Finally, Section 6 concludes the paper and recommends future work in this area of study.

2. Related Work

Sentiment analysis (SA) is a branch in NLP, which utilizes text mining and related technologies to classify subjective text into classes of opinions, emotions, or any other category. Vector representation of text is a very important task in sentiment analysis since it determines the accuracy and efficiency of the developed SA models [5]. Recently, there are many studies that have used lexicon-based techniques, pre-trained word embeddings, NLP techniques, and deep learning models in vector representation and generally in SA. In Section 2.1, current research in lexicon-based techniques, N-grams, and NLP is discussed, whereas in Section 2.2, research in pre-trained word embeddings and deep learning models is discussed.

2.1. Lexicon-Based Techniques, N-Grams and Natural Language Processing

The lexicon-based techniques use a dictionary of words labeled with their sentiment orientations. In such techniques, a piece of text is converted into a bag of words whose sentiment orientations are summarized or aggregated to classify the text. This technique is simple, but it is mostly dependent on manual labeling of the text [22]. Baharudin and Khan [23] suggested that sentence structure and contextual information are important for sentiment orientation and classification. In their work, each term in the sentence was assigned a sentiment score from the Sent WordNet lexicon. The overall classification of the sentence is the sum total score of the individual scores of the terms in the sentence. While the approach is interesting, one of the limitations of this approach is that words can be of the same orientation, but negating one another, thus giving the wrong sentiment classification. The main improvements of lexicon-based techniques involve using lexicon labeled words as input to machine learning classifiers. Mudinas et al. [24] combined lexicon-based approach and support vector machine. In their method, they generated word sentiment labels and used them as input to the SVM classifier. Seyed et al. [14] used several lexicons to assign lexicon vectors to words in a text, which they referred to as Lexicon2Vec (L2V). They combined their vector with Word2Vec and PoS2vec to obtain a hybrid vector representation. Generally, little research has been performed on combining lexicon-based methods and deep learning architectures. Huang et al. [25] proposed a sentiment analysis model of online reviews, which they referred to as polymerization topic sentiment model (PTSM). In their model, they used lexicon dictionary to extract sentiment information from online reviews. Although their model performed well with the support vector machine, they did not test their model with deep learning classifiers or word embedding algorithms. However, they recommended use of lexicon-based methods to solve the over-fitting problem of sentiment analysis models and to filter unnecessary information

Generation of word N-grams is another important NLP technique applicable in sentiment analysis. In text classification, word-grams are used to generate word co-occurrence patterns and vectors for machine learning classifiers. N-gram NLP models are widely used due to their simplicity and effectiveness [26]. However, they do not consider the information encapsulated in the sequence of the words. For instance, words could be negating one another in a sentence or having different meaning in different context. Kumar et al. [27], in their recent research on use of N-grams in text representation, used bi-grams and tri-grams to extract features from text data. Their work yielded promising results, which is an indication that N-grams can be utilized for effective text representation. They proposed a big data analytics framework for sentiment analysis and classification using intelligent cognitive inspired computing. In their model, they used fuzzy cognitive maps as classifiers. In our research, we advance this work by investigating use of hybrid NLP techniques, including N-grams and sentiment lexicon. They also recommended future research on deep learning architecture, an area which is also being explored in this research work. We do so by seeking to combine pre-trained word embeddings with sentiment lexicon and N-grams.

2.2. Word Embeddings-Based Techniques and Deep Learning Models

Recently, word embeddings-based vector representation techniques are playing an important role in natural language processing [28]. According to Mikolov [29], research in word embedding feature selection gained momentum in 2013. The main word embeddings algorithms are Word2Vec [15], Glove [16] and FastText [17,30], which are used to convert words to vectors. Recently, bidirectional encoder representations from the transformers (BERT) model [18] has received much attention due to its bidirectional and attention mechanisms. Consequently, use of BERT embedding-based models outperforms other models, thus showing remarkable performance in sentiment analysis tasks [31,32]. Word embeddings are better than the normal bag of words representation, since they cater for synonyms and produce vectors with lower dimensionality than the bag of words [14,15]. Garg [33] did research on word embeddings and established that Word2Vec embeddings performed better than the other word-embedding algorithms. Currently, most researchers use pre-trained word embeddings vectors as inputs of machine learning classifiers in their sentiment analysis research since they are more accurate and compatible with deep learning neural networks [22]. However, pre-trained word embeddings ignore sentiment orientation of words and their context, hence affecting sentiment classification accuracy [14,28]. This is because they use word distances and synonyms to calculate word vectors.

Kim [34] studied use of pre-trained Word2Vec vectors as inputs to convolutional neural networks and improved their performance by hyper parameter tuning of the CNN model. Wang et al. [35] used pre-trained Glove vectors as inputs for attention-based LSTM models for aspect-level sentiment analysis. Liu et al. [21] used pre-trained Word2Vec in idiom recommendation model in essay writing. Liu et al. [36] used pre trained Word2Vec model and improved them for cross-domain classification by extending the vector to include domain information. Recently D'Silva and Sharma [37] used FastText pre-trained word embeddings and neural networks to classify Konkani texts. Hu et al. [38] used BERT to integrate mental features and short text vector to improve topic classification and false detection in short text. Although their work showed better performance, they did not compare their proposal with other word embedding models. They also suggested more research to be performed on application of BERT in other contexts of text classification. Prottasha et al. [31] did a study to compare Word2Vec, Glove, FastText, and BERT. They demonstrated that transformer architectures, such as BERT models, are the state-of-the-art models for text representation and play a crucial role in sentiment analysis. The superiority of BERT is that it can read series of words in either direction, unlike other word embedding algorithms. Further, BERT employs the attention mechanism of the transformer that assigns a word its vector, depending on the surrounding words. This mechanism enhances the semantic representation of the target text. However, the series of input words to be read by the BERT algorithm maintains the entire words of the target text. We propose that the performance of BERT algorithm can be enhanced by focusing the input series to a few words, which contain sentiment information and their neighbours of the target text. This can be guided by utilization of sentiment lexicon and word N-grams. In a recent study [13], the researchers investigated a text representation technique using sentiment lexicon and N-grams where a Lexicon-pointed hybrid N-gram feature extraction model (LeNFEM) was proposed and investigated. A three-word N-gram was identified, which contains a sentiment word by use of a sentiment lexicon. The N-gram was then expanded to form a hybrid vector containing words, POS tags, and sentiments. Although this is a novel text representation technique, a proposal was put forth on investigation of how the approach could be applied with deep learning models, including word embeddings. In this paper, we extend on this work and present a text representation technique named lexicon selected-BERT embedding (LeBERT) Model. The model combines sentiment lexicon and BERT word embeddings via word N-grams for sentiment classification.

Based on the related work discussed, we observe that existing deep learning models for sentiment analysis generate text representation vectors using word embeddings. We also noted that the BERT model is one of the state-of-the-art embedding models. Thus,

any study on improving it advances sentiment analysis and natural language processing research. With this objective, this study suggested and investigated combination of BERT word embedding model, sentiment lexicon, and N-grams. The novelty of the proposed LeBERT model is that the sentiment lexicon is utilized to identify a section of a text (sentence or a document) where sentiment information is domiciled, and the BERT algorithm is used to build word vectors from that section only. In Section 3, we present and describe the details of the proposed model.

3. The Proposed LeBERT Model

In deep learning, the BERT model is one of the current word embeddings and text representation models under study for sentiment analysis. BERT, unlike other word embedding algorithms, can effectively read series of words in either direction of the input text, and since it uses the attention mechanism to assign a word, its vector depends on the surrounding words, and it is efficient in word vectorization [39]. Although BERT considers the context of a word when assigning the vector, it does so for all the words in the input text, which leads to a resultant vector with high dimensionality. Second, word vectors built from BERT do not contain semantic information, which is critical in sentiment classification. Compared with BERT, the sentiment lexicon can be used to identify sentiment words in a text and assign specific sentiment polarity to the words. However, sentiment lexicon cannot generate representative word vectors, hence leading to high data sparseness. Thus, to improve sentiment classification, this paper proposes the LeBERT model, which combines sentiment lexicon, N-grams, and BERT algorithms.

The design idea of the LeBERT model is to first use N-grams to split the input text into sections, and then use a sentiment lexicon to identify a section or sections that contain a sentiment word. It is worth noting that text reviews, such as social media posts, contain short text, and characteristically, semantic features in short texts are concentrated in a certain part [39]. Thus, extracting features from such parts will lead to efficient and effective text representation. The words of the identified section(s) are then converted into a vector by BERT. The output word vector is then used as the input into a CNN model with a fully connected layer where features from the vector are obtained. The features extracted are then integrated by the dense output layer, and finally the sentiment class of the text is performed by a SoftMax classifier. The architecture of the proposed LeBERT model is shown in Figure 1.

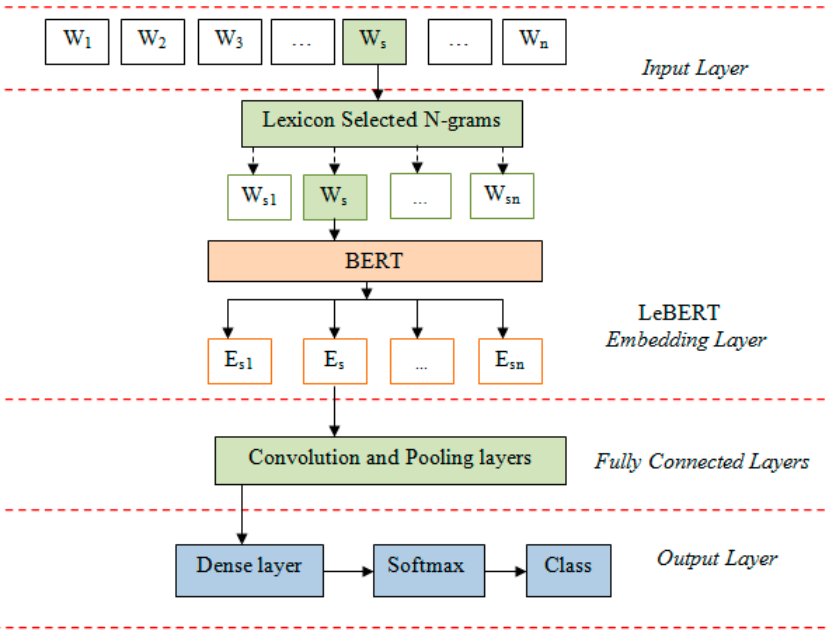

Figure 1. Architecture of the proposed sentiment analysis model.

As shown in Figure 1, the sentiment lexicon, N-grams, and BERT algorithm are used in the embedding layer to build the word vector. The overall sentiment analysis model using the LeBERT model is presented in Figure 2.

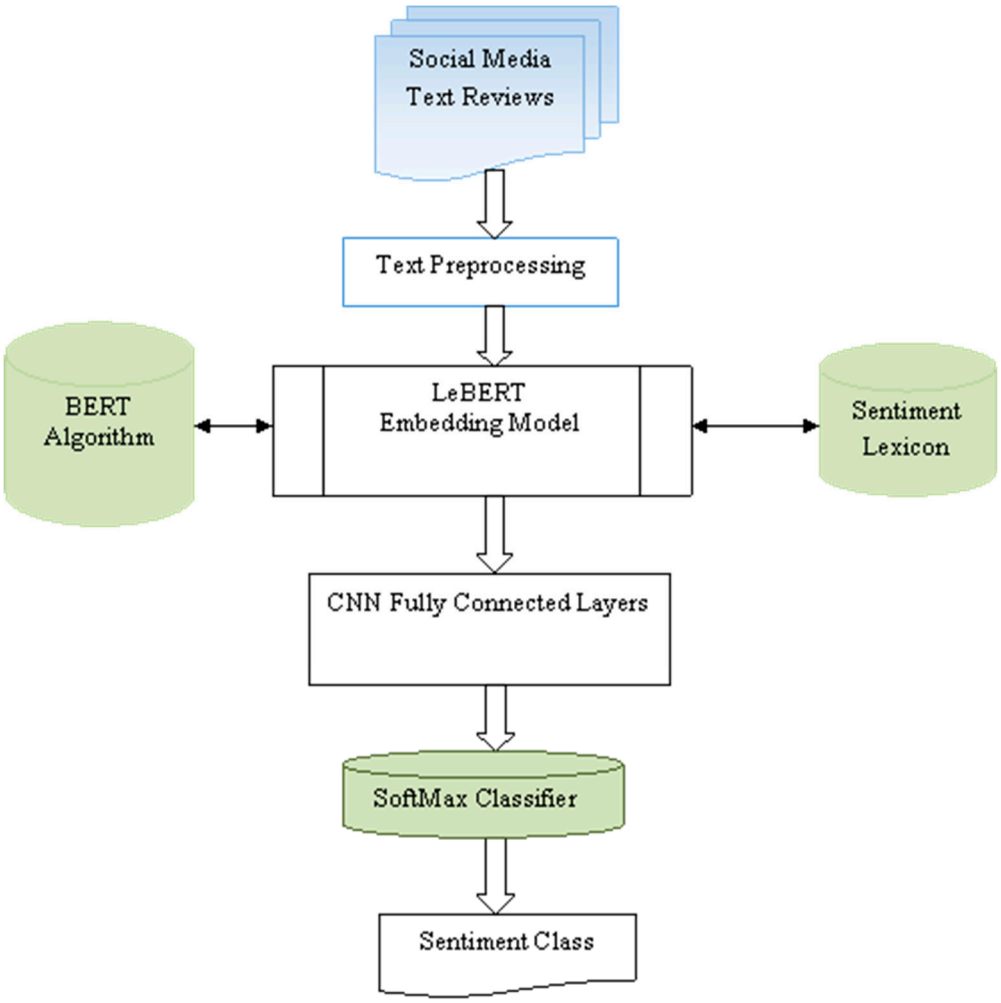

Figure 2. Sentiment analysis model using the LeBERT model.

3.1. LeBERT Embedding

There are currently two common methods used to build text vectors for sentiment analysis: word-embedding based methods or lexicon-based methods. In our proposed model, we sought to utilize both methods through N-grams. The sentiment lexicon is used to identify word N-grams containing a sentiment word, and then the vector from the N-gram words using BERT word embedding model is used.

To build the vector, we first generate word N-grams from the sentences. A N-gram is a combination of words from a sentence, which forms a Markovian process. Normally, this is used to predict the next word in a sequence of words. Further, Markovian process also generates co-occurrence of words, which is a key aspect in influencing sentiment in a text. In this case, we use N-gram sequences to partition a sentence into various sections that represent the entire text, such as an online review or a sentence. This is because N-grams present co-occurrence of words in a text in a more comprehensive manner than mere bag of words (BoW). The size of the partition depends on the value of N.

For instance, if we consider a sentence S given as:

$$S = \{w1, w2, w3, w4, w5, \ldots, \ldots, \ldots wn\} \quad (1)$$

where, w_i are words.

For various values of N, we have;

N = 1, the set of N-grams $N_1 = \{w_1, w_2, w_3, \ldots, w_n\}$

N = 2, the set of N-grams $N_2 = \{w_1\_w_2, w_2\_w_3, w_3\_w_4, \ldots, w_{n-1}\_w_n\}$

N = 3, the set of N-grams $N_3 = \{w_1\_w_2\_w_3, w_2\_w_3\_w_4, w_3\_w_4\_w_5, \ldots, w_{n-2}\_w_{n-1}\_w_n\}$

The fundamental idea is that, with the set of N-grams, it is possible to select a section of the entire input text. This ensures that we use the most significant words when building text vectors for sentiment analysis. Once the N-gram(s) are identified from the text, it is then reverted to a bag of words. Each word is then converted into a vector using the BERT word-embedding algorithm.

3.2. The LeBERT Embedding Algorithm

Let L: sentiment lexicon; C: corpus of subjective user reviews (R_i); $\boldsymbol{V_i}$: vector representation of a subjective review (R_i); W_t: sentiment term; W_1: the first word neighboring the sentiment term; and W_2: the second word neighboring the sentiment term.

We define the text vector, $\boldsymbol{v_i}$, of a subjective review, R_i, as the vector originating from a selected section of the review S_i using sentiment lexicon and BERT word embedding model (***Be***). The algorithm listing of the sentence vector representation generation is presented in Algorithm 1.

Algorithm 1 Contextualized Text Vector Generation

Inputs:
$R_i = \{w_1, w_2, \ldots \ldots, w_n)$, input review containing n words
L = sentiment lexicon
Be = BERT word-embedding model
Output: Contextualized Text Vector (v_i), representing the subjective user review
START
Set the N-gram value to N = 3
FOR each review ($R_i \epsilon C$) with n word tokens
PRINT the word trigrams;
Call the sentiment lexicon (L)
FOR each trigram check for a sentiment word;
IF a trigram contains a sentiment word **THEN**
PRINT the trigram words (w_1, w_t, w_2)
ELSE delete the trigram
ENDIF
END
Generate section vector(·)
FOR Each word (w_1, w_t, and w_2) in the trigram
READ (w_i) into gag of words ($\boldsymbol{B_{wi}}$)
Call the pre-trained word-embedding (***Be***)
Calculate the word vector ($\boldsymbol{w_{vi}}$)
END
Update vector$\boldsymbol{V_i}$:‹$\boldsymbol{w_{v1}}$ **and** $\boldsymbol{w_{vt}}$ **and** $\boldsymbol{w_{v2}}$›
END
Return Vector$\boldsymbol{V_i}$.

3.3. The CNN Layer

The CNN deep learning model is used as the classifier, which uses the resultant vector from LeBERT embedding as input and gives the sentiment class as the output. CNNs are specialized types of artificial neural networks, which are capable of outperforming the common machine learning algorithms in supervised learning tasks. CNNs' main function is to identify and learn the information characteristic patterns through the use of convolution layers and thus facilitate classification of the objects. The CNN model is presented in Figure 3. Using the convolution kernels (windows) and the nonlinear function (filter), feature maps are obtained. A pooling operation is then applied on the feature maps to

select the optimal features. The dense output layer then classifies the optimal features using softmax activation function (which uses probability) into a positive or a negative class.

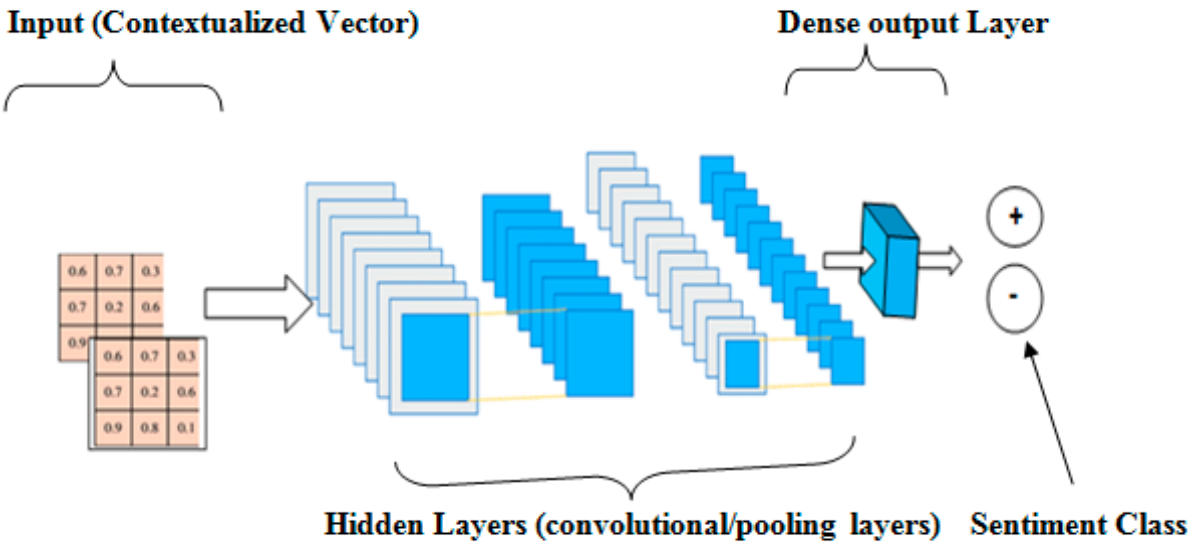

Figure 3. The CNN model.

4. Experiments

This section describes the dataset used; the experiments set up was carried out to evaluate the performance of the proposed model. The tools and techniques used in model formulation and evaluation are also discussed.

4.1. Dataset

In order to evaluate the effectiveness of the proposed model, the experiments were carried using a dataset complied from three public datasets. The dataset contains three world datasets including: Amazon products' reviews dataset, with 70,000 reviews, Imbd dataset, with 50,000 movie reviews, and Yelp dataset, with 300,000 restaurants' reviews. In the experiments, we used 3000 reviews, as compiled by Kotzias et al. [40] and published in a machine learning repository. For each website, Kotzias et al. [40] randomly sampled 500 positive and 500 negative tweets, which were clearly positive and negative.

4.2. Experiment Setup

The reviews presented in the dataset were cleaned of non-English words and pre-processed. Tokenization, N-grams generation, text vector building, and designing of the CNN layers was conducted using python programming language in the virtual laboratory (Google Colaboratory). The obtained vector was split into two subsets, 80% of the dataset was used for training the CNN model, and the other 20% was used for evaluating the classification performance. Since the dataset contained multiple sentences (reviews), pooled output was used in the BERT embedding. The rectified linear unit (RELU) was used as the activation function, with 100 neurons for the hidden fully connected layer. The output dense layer was set up with two (2) neurons since the texts were to be classified into two classes. Softmax was used as the activation function, which was in line with the text classification problem at hand. In the study, we used 50-dimensional Glove word embeddings trained on Google News, 250-dimensional Word2Vec embeddings trained on Wikipedia, and 128-dimensional BERT embeddings trained on English Wikipedia corpus. In the experiments, we used tensor flow tools to prepare the data and build our proposed model. Among the training set, a small potion (100) of the reviews was used for validation. In Section 4.3, we present the model parameters of the designed CNN model.

4.3. Model Parameters BERT, Glove, and Word2Vec Pre-Trained Word Embeddings

The model parameters for the BERT word embeddings were as shown in Table 1.

Table 1. Model Parameters for BERT word embedding.

Layer (Type)	Output Shape	Parameters
Keras Layer	(None, 128)	4,385,921
Dense Hidden layers	(None, 16)	2064
Dense Output Layer	(None, 1)	17
Total Parameters 4,388,002 Trainable Parameters: 4,388,001 Non-trainable Parameters: 1		

From Table 1, the Keras layer represents the shape of embedding and the preprocessor used for the BERT model. In the experiment, the BERT word embeddings were initialized using small BERT due to limitations of computation resources. Consequently, the dimension of the word embedding was set to 128 and appropriate preprocessor for the BERT was set. Glove and Word2Vec word embeddings of 50 and 250 dimensions, respectively, were used as baseline models, and their parameters were set as shown in Tables 2 and 3.

Table 2. Model Parameters for 50-diemsional Glove word embeddings.

Layer (Type)	Output Shape	Parameters
Keras Layer	(None, 50)	48,190,600
Dense Hidden layers	(None, 16)	816
Dense Output Layer	(None, 1)	17
Total Parameters 48,191,433 Trainable Parameters: 48,191,433 Non-trainable Parameters: 0		

Table 3. Model Parameters for 250-dimensional Word2Vec word embeddings.

Layer (Type)	Output Shape	Parameters
Keras Layer	(None, 250)	252,343,750
Dense Hidden layers	(None, 16)	4016
Dense Output Layer	(None, 1)	17
Total Parameters 48,191,433 Trainable Parameters: 48,191,433 Non-trainable Parameters: 0		

From Tables 2 and 3, The Keras layer represents the input layer in which the input vector was obtained using the Glove and Word2Vec word embeddings. The shape of the Keras layer was determined by the dimensions of the word embeddings. The dense output layer is for binary classification of the input text into positive or negative sentiment.

4.4. Model Performance Evaluation

To verify the effectiveness of the proposed model, a 2 by 2 contingency matrix that shows the number of correctly predicted positive reviews (TP), true negative reviews (TN), false positive reviews (FP), and false negative reviews [41] was used, as shown in Table 4.

Table 4. Contingency Table.

	Classified as Positive	**Classified as Negative**
Actual Positive	TP	FN
Actual Negative	FP	TN

Four model evaluation metrics were selected: accuracy, precision, recall, and F-measure. From Table 4, we calculated the metrics, as discussed and presented in Equations (2)–(5).

Accuracy is the ratio of the correctly classified predictions to the total sum of predictions. It is given as;

$$Accuracy = \frac{TP + TN}{(TP + FN + FP + TN)} \quad (2)$$

Precision is the ratio of accurately classified data to the total data classified in the class. It is given as;

$$Precision = \frac{TP}{(TP + FP)} \quad (3)$$

Recall is the ratio of accurately classified data to the actual data in the class. It is given as;

$$Recall = \frac{TP}{(TP + FN)} \quad (4)$$

F-measure is the mean of precision and recall. It is given as;

$$\text{F} - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{(\text{Precision} + \text{Recall})} \quad (5)$$

5. Results and Discussion

This section describes the results obtained from the experiments. We first sought to test the effect of using sentiment lexicon on the input text data and the vector. We compared the shape of Yelp dataset (restaurants reviews) before and after using the sentiment lexicon. Table 5 presents the details of the text data.

Table 5. Details of the text data before and after using sentiment lexicon.

Text Data Item	**Before Using the Sentiment Lexicon**	**After Using the Sentiment Lexicon**
Characters(no spaces)	46,744	14,182
Characters(with spaces)	56,616	19,212
Number of words	10,863	2989
Number of paragraphs	996	996
Average Number of words per Post/paragraph	11	3

From Table 5, it was evident that application of sentiment lexicon to select a section of the input text significantly reduced the size of input text. Although the number of posts or paragraphs remained the same, the shape of the input text changed from 11 rows to 3 rows, which, in turn, would reduce the computation time for the model. We then designed and performed experiments with deep learning CNN to evaluate how the LeBERT embedding model would perform in sentiment analysis.

5.1. Ablation Study on Effect of Size of N-Grams on LeBERT Model

In order to verify the effectiveness of using LeBERT model as the embedding layer to generate word vectors, we first did an experiment to study the effect of the size of N-grams on the LeBERT model with CNN. In the experiment, the restaurant reviews datasets were used. The experimental results of N = 1,2, 3 and all words were as shown in Table 6.

Table 6. Sentiment classification results of various sizes of N-grams with the LeBERT model.

N-Grams	LeBERT-CNN			
	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
N = 1	65.02	65.02	65.15	65.08
N = 2	79.45	79.50	80.04	79.77
N = 3	88.20	88.45	89.01	**88.73**
N = 4	87.65	87.65	87.80	87.72
All words	84.00	84.00	84.20	84.10

For N = 1, it implies that, for each sentence, only one word was used, which was chosen by the sentiment lexicon. The results indicate a low performance since one word cannot represent the sentiment of the entire text. The highest model performance was obtained at N = 3. As shown in Table 6, we generated N-grams up to N = 4 due to computational resources. The category of 'All words' implies that the sentiment lexicon was not applied on the input text to select some words, hence, this reverts to the original BERT model. The results indicated that N = 3 is an ideal size of N-gram for the proposed model. Section 5.2 presents the performance results of the model in comparison to the baseline models in the three datasets.

5.2. Comparison of LeBERT Model Performance with Baseline Models

The experiment was carried out to validate the performance of the proposed LeBERT model in terms of accuracy, recall, precision, and F-measure of the CNN on the three discussed datasets. Glove and Word2Vec were used as baseline word embedding models. In this experiment, tri-grams (N = 3) were used. Tables 7–9 show the performance results on restaurants reviews, movie reviews, and product reviews datasets, respectively.

Table 7. Sentiment classification prediction under Yelp dataset (restaurant reviews).

Embedding Model	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
Glove	78.50	78.56	78.70	78.63
Le-Glove	81.50	82.00	83.01	82.50
Word2Vec	75.50	75.50	75.80	75.65
Le-Word2Vec	82.40	82.45	83.15	82.80
BERT	84.00	84.00	84.20	84.10
LeBERT(our Model)	88.20	88.45	89.01	**88.73**

Table 8. Sentiment classification prediction under IMDB dataset (movie reviews).

Embedding Model	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
Glove	79.50	79.50	80.10	79.80
Le-Glove	82.50	82.70	83.25	82.97
Word2Vec	77.45	77.46	78.01	77.73
LeWord2Vec	83.00	83.02	83.42	83.22
BERT	84.01	84.08	84.63	84.35
LeBERT (our Model)	86.10	86.71	87.00	**86.85**

The presented tables indicate the comparative results between the pre-trained word embeddings, with and without the proposed fusion with sentiment lexicon. Generally, the proposed LeBERT model performs better compared to the baseline word embeddings models. Accuracy is considered to be a good performance evaluation metric when the classes are balanced [41]. Since, in our experiments all the three datasets exhibited balanced classes, we compared accuracy of the model with the various approaches for the three datasets. The results obtained were as shown in Figure 4.

Table 9. Sentiment classification prediction under Amazon dataset (products reviews).

Embedding Model	Accuracy (%)	Precision (%)	Recall (%)	F-Measure (%)
Glove	79.00	79.00	79.65	79.32
Le-Glove	79.60	80.00	80.45	80.22
Word2Vec	79.50	79.50	80.25	79.87
Le-Word2Vec	81.50	81.50	82.05	81.77
BERT	81.72	81.75	82.04	81.89
LeBERT	82.40	82.40	82.64	**82.52**

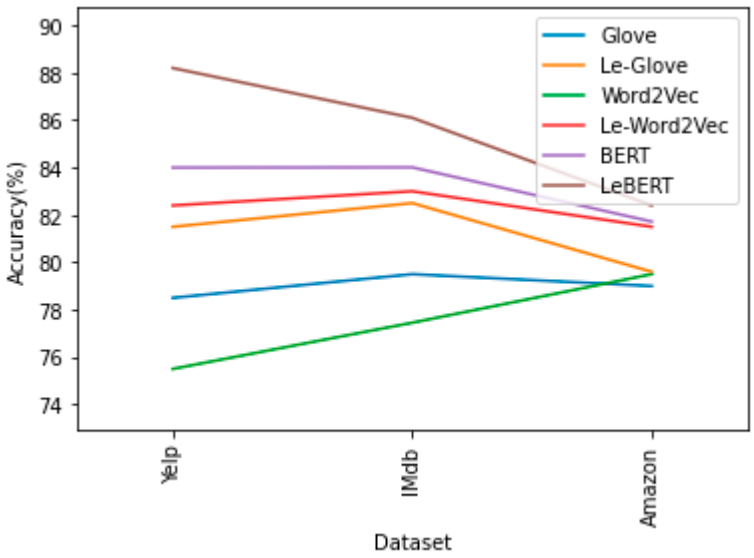

Figure 4. Sentiment prediction accuracy using various embedding models.

From Figure 4, our proposed model (LeBERT) had the highest accuracy in all datasets, with relatively lower accuracy on Amazon's product reviews dataset. This could be attributed to the fact that the reviews referred to various products, and thus the sentiment terms varied from one product to another.

6. Conclusions

Sentiment analysis of social media reviews is a difficult task due to sparsity and high dimensionality of word vectors representing the text. Use of sentiment lexicon and word embedding algorithms can improve sentiment analysis models for text reviews. In this context, we proposed a sentiment analysis model, named LeBERT, based on sentiment lexicon, N-grams, BERT word embedding, and CNN. In the model, a section of a document or a sentence where sentiment information can be highly found is selected using sentiment lexicon and word N-grams, and then the words are vectorized using the BERT word embedding algorithm. A CNN classifier is then used to classify the input vector into a sentiment class. To validate the performance of the proposed LeBERT model, original Word2Vec, Glove, and BERT word embeddings were used as baseline models on three benchmark sentiment datasets. From the experimental results, use of sentiment lexicon significantly reduces the dimension of the input vector, thus improving efficiency of sentiment analysis models. Secondly, integration of sentiment lexicon and N-grams with BERT embedding algorithm yields a better representative word vector, hence increasing the predictive performance of the resultant sentiment analysis model. The results also indicated that sentiment lexicon with BERT (through LeBERT model) outperformed other word embedding algorithms.

This paper had some limitations. The designed model utilized convolutional neural network (CNN) only. In the future, the LeBERT embedding model could be implemented and evaluated in other neural networks, such as long short-term memory (LSTM). Our

proposed model was tested and found to be effective in binary sentiment classification, where sentiment lexicon was used. It would be interesting to evaluate the model on other text classification tasks where other types of lexicons are used.

Author Contributions: Conceptualization, J.M., W.M. and G.O.; Data curation, J.M.; Formal analysis, G.O. and W.M.; Investigation, J.M., G.O. and W.M.; Methodology, J.M., G.O. and W.M.; Supervision, G.O. and W.M.; Writing—original draft, J.M.; Writing—review and editing, G.O. and W.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in the experiments is publicly available from https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences accessed on 6 October 2022. The dataset was compiled by Kotzias et al. [40], which is cited in this research work.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Jiang, Z.; Gao, B.; He, Y.; Han, Y.; Doyle, P.; Zhu, Q. Text Classification Using Novel Term Weighting Scheme-Based Improved TF-IDF for Internet Media Reports. *Math. Probl. Eng.* **2021**, *2021*, 6619088. [CrossRef]
2. Onan, A.; Üniversitesi, I.K. Ensemble of Classifiers and Term Weighting Schemes for Sentiment Analysis in Turkish. *Sci. Res. Commun.* **2021**. [CrossRef]
3. Kalarani, P.; Selva, B.S. An overview on research challenges in opinion mining and sentiment analysis. *Int. J. Innov. Res. Comput. Commun. Eng.* **2015**, *3*, 1–6.
4. Yang, J.; Xiu, P.; Sun, L.; Ying, L.; Muthu, B. Social media data analytics for business decision making system to competitive analysis. *Inf. Process. Manag.* **2021**, *59*, 102751. [CrossRef]
5. Rao, L. Sentiment Analysis of English Text with Multilevel Features. *Sci. Program.* **2022**. [CrossRef]
6. Onan, A.; Korukoğlu, S. A feature selection model based on genetic rank aggregation for text sentiment classification. *J. Inf. Sci.* **2016**, *43*, 25–38. [CrossRef]
7. Bhadane, C.; Dalal, H.; Doshi, H. Sentiment Analysis: Measuring Opinions. *Procedia Comput. Sci.* **2015**, *45*, 808–814. [CrossRef]
8. Mozetič, I.; Grčar, M.; Smailović, J. Multilingual Twitter Sentiment Classification: The Role of Human Annotators. *PLoS ONE* **2016**, *11*, e0155036. [CrossRef]
9. Li, B.; Guoyong, Y. Improvement of TF-IDF Algorithm based on Hadoop Framework. In Proceedings of the 2nd International Conference on Computer Application and System Modeling, Taiyuan, China, 27–29 July 2012; pp. 391–393.
10. Ankit, N.S. An Ensemble Classification System for Twitter Sentiment Analysis. *Procedia Comput. Sci.* **2018**, *132*, 937–946. [CrossRef]
11. Ahuja, R.; Chug, A.; Kohli, S.; Gupta, S.; Ahuja, P. The Impact of Features Extraction on the Sentiment Analysis. *Procedia Comput. Sci.* **2019**, *152*, 341–348. [CrossRef]
12. Rao, G.; Huang, Z.F.; Cong, Q. LSTM with sentence representations for document level sentiment classification. *Neurocomputing* **2018**, *308*, 49–57. [CrossRef]
13. Mutinda, J.; Mwangi, W.; Okeyo, G. Lexicon-pointed hybrid N-gram Features Extraction Model (LeNFEM) for sentence level sentiment analysis. *Eng. Rep.* **2021**, *3*, e12374. [CrossRef]
14. Rezaeinia, S.M.; Rahmani, R.; Ghodsi, A.; Veisi, H. Sentiment analysis based on improved pre-trained word embeddings. *Expert Syst. Appl.* **2019**, *117*, 139–147. [CrossRef]
15. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781. [CrossRef]
16. Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 26–28 October 2014; pp. 1532–1543.
17. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
18. Kenton, J.D.M.W.C.; Toutanova, L.K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the NaacL-HLT, Minneapolis, Minnesota, 2 June 2019; pp. 4171–4186.
19. Sharma, A.K.; Chaurasiaa, S.; Srivastavaa, D.K. Sentimental Short Sentences Classification by Using CNN Deep Learning Model with Fine Tuned Word2Vec. *Procedia Comput. Sci.* **2020**, *167*, 1139–1147. [CrossRef]
20. Dashtipour, K.; Gogate, M.; Adeel, A.; Larijani, H.; Hussain, A. Sentiment Analysis of Persian Movie Reviews Using Deep Learning. *Entropy* **2021**, *23*, 596. [CrossRef]

21. Liu, Y.; Liu, B.; Shan, L.; Wang, X. Modelling context with neural networks for recommending idioms in essay writing. *Neurocomputing* **2018**, *275*, 2287–2293. [CrossRef]
22. Giatsoglou, M.; Vozalis, M.G.; Diamantaras, K.; Vakali, A.; Sarigiannidis, G.; Chatzisavvas, K.C. Sentiment analysis leveraging emotions and word embeddings. *Expert Syst. Appl.* **2017**, *69*, 214–224. [CrossRef]
23. Baharudin, B.; Khan, A. Sentiment Classification Using Sentence-level Semantic Orientation of Opinion Terms from Blogs. In Proceedings of the 2011 National Postgraduate Conference, Perak, Malaysia, 19–20 September 2011; IEEE: Piscataway, NJ, USA, 2011. [CrossRef]
24. Mudinas, A.; Zhang, D.; Levene, M. Combining lexicon and learning based approaches for concept-level sentiment analysis. In Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining, Beijing, China, 12 August 2012; pp. 1–8.
25. Huang, L.; Dou, Z.; Hu, Y.; Huang, R. Textual Analysis for Online Reviews: A Polymerization Topic Sentiment Model. *IEEE Access* **2019**, *7*, 91940–91945. [CrossRef]
26. Fotis, A.; Dimitrios, T.; John, V.; Theodora, V. Using N-Gram Graphs for Sentiment Analysis: An Extended Study on Twitter. In Proceedings of the 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService), Oxford, UK, 29 March–1 April 2016; pp. 44–51.
27. Jain, D.K.; Boyapati, P.; Venkatesh, J.; Prakash, M. An Intelligent Cognitive-Inspired Computing with Big Data Analytics Framework for Sentiment Analysis and Classification. *Inf. Process. Manag.* **2022**, *59*, 102758. [CrossRef]
28. Araque, O.; Corcuera-Platas, I.; Sánchez-Rada, J.; Iglesias, C. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Syst. Appl.* **2017**, *77*, 236–246. [CrossRef]
29. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, CA, USA, 5–10 December 2013; pp. 3111–3119.
30. Chandrasekaran, G.; Nguyen, T.N.; Hemanth, J.H. Multimodal sentimental analysis for social media applications: A comprehensive review. WIREs Data Min. *Knowl. Discov.* **2021**, *11*, e1415.
31. Prottasha, N.J.; Sami, A.A.; Kowsher; Murad, S.A.; Bairagi, A.K.; Masud, M.; Baz, M. Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning. *Sensors* **2022**, *22*, 4157. [CrossRef] [PubMed]
32. Jain, P.K.; Quamer, W.; Saravanan, V.; Pamula, R. Employing BERT-DCNN with sentic knowledge base for social media sentiment analysis. *J. Ambient. Intell. Humaniz. Comput.* **2022**, 1–13. [CrossRef]
33. Garg, S.B.; Subrahmanyam, V.V. Sentiment Analysis: Choosing the Right Word Embedding for Deep Learning Model. In *Advanced Computing and Intelligent Technologies*; Lecture Notes in Networks and Systems; Bianchini, M., Piuri, V., Das, S., Shaw, R.N., Eds.; Springer: Singapore, 2022; Volume 218. [CrossRef]
34. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 26–28 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1746–1751.
35. Wang, Y.; Huang, M.; Zhu, X.; Zhao, L. Attention-based LSTM for Aspect-level Sentiment Classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, 1–5 November 2016; pp. 606–615.
36. Liu, J.; Zheng, S.; Xu, G.; Lin, M. Cross-domain sentiment aware word embeddings for review sentiment analysis. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 343–354. [CrossRef]
37. D'Silva, J.; Sharma, U. Automatic text summarization of konkani texts using pre-trained word embeddings and deep learning. *Int. J. Electr. Comput. Eng. (IJECE)* **2022**, *12*, 1990–2000. [CrossRef]
38. Hu, Y.; Ding, J.; Dou, Z.; Chang, H. Short-Text Classification Detector: A Bert-Based Mental Approach. *Comput. Intell. Neurosci.* **2022**. [CrossRef]
39. Yang, H. Network Public Opinion Risk Prediction and Judgment Based on Deep Learning: A Model of Text Sentiment Analysis. *Comput. Intell. Neurosci.* **2022**, *2022*. [CrossRef]
40. Kotzias, D.; Denil, M.; de Freitas, N.; Smyth, P. From Group to Individual Labels Using Deep Features. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 15 August 2015; Association for Computing Machinery: New York, NY, USA; pp. 597–606. [CrossRef]
41. Singh, K.N.; Devi, S.D.; Devi, H.M.; Mahanta, A.K. A novel approach for dimension reduction using word embedding: An enhanced text classification approach. *Int. J. Inf. Manag. Data Insights* **2022**, *2*, 100061. [CrossRef]