

Article

Document-Level Sentiment Analysis Using Attention-Based Bi-Directional Long Short-Term Memory Network and Two-Dimensional Convolutional Neural Network

Yanying Mao ^{1,2}, Yu Zhang ^{3,4} , Liudan Jiao ^{3,*} and Heshan Zhang ³

¹ Department of Communication Engineering, Chongqing College of Electronic Engineering, Chongqing 401331, China; maoyanying@foxmail.com

² College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

³ School of Economics and Management, Chongqing Jiaotong University, Chongqing 400074, China; yuzhang@cqjtu.edu.cn (Y.Z.); hszhang@cqjtu.edu.cn (H.Z.)

⁴ T. Y. Lin International Engineering Consulting (China) Co., Ltd., Chongqing 401121, China

* Correspondence: jld@cqjtu.edu.cn

Abstract: Due to outstanding feature extraction ability, neural networks have recently achieved great success in sentiment analysis. However, one of the remaining challenges of sentiment analysis is to model long texts to consider the intrinsic relations between two sentences in the semantic meaning of a document. Moreover, most existing methods are not powerful enough to differentiate the importance of different document features. To address these problems, this paper proposes a new neural network model: AttBiLSTM-2DCNN, which entails two perspectives. First, a two-layer, bidirectional long short-term memory (BiLSTM) network is utilized to obtain the sentiment semantics of a document. The first BiLSTM layer learns the sentiment semantic representation from both directions of a sentence, and the second BiLSTM layer is used to encode the intrinsic relations of sentences into the document matrix representation with a feature dimension and a time-step dimension. Second, a two-dimensional convolutional neural network (2DCNN) is employed to obtain more sentiment dependencies between two sentences. Third, we utilize a two-layer attention mechanism to distinguish the importance of words and sentences in the document. Last, to validate the model, we perform an experiment on two public review datasets that are derived from Yelp2015 and IMDB. Accuracy, F1-Measure, and MSE are used as evaluation metrics. The experimental results show that our model can not only capture sentimental relations but also outperform certain state-of-the-art models.

Keywords: sentiment analysis; bidirectional LSTM; 2DCNN; attention mechanism



Citation: Mao, Y.; Zhang, Y.; Jiao, L.; Zhang, H. Document-Level Sentiment Analysis Using Attention-Based Bi-Directional Long Short-Term Memory Network and Two-Dimensional Convolutional Neural Network. *Electronics* **2022**, *11*, 1906. <https://doi.org/10.3390/electronics11121906>

Academic Editors: Matei Mancas and Jean-Benoit Delbrouck

Received: 4 June 2022

Accepted: 16 June 2022

Published: 18 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Sentiment analysis is an essential research topic in natural language processing (NLP). It can be used to analyse the emotional tendencies of people by classifying reviews and opinions regarding products or services [1]. It has been widely employed in economic analysis, online social networks, and other areas. Text sentiment analysis is important and interesting research as it can help companies and sellers understand what users and buyers feel about their products and services and whether consumers like their products and services. Thus, it can help users and buyers identify business opportunities [2]. Moreover, the analysis of the comments and opinions of internet users helps governments control the development of social events and understand public opinion so that corresponding measures can be taken [3,4]. Document-level comment text exists on various platforms, including e-commerce, forums, blogs, etc., with a wide range of platforms and obvious positive and negative emotional tendencies [5]. Therefore, the analysis of the sentiment orientation of document-level comment text is highly practical.

The sentiment analysis of text can be divided into three categories, namely, the phrase level, sentence level, and document level [6]. According to the sentiment tendency of features, sentiment classification can easily be performed on the phrase and sentence levels [7]. However, it is difficult to classify document level datasets as the sentiment semantics and the dependency between two sentences need to be considered [8]; therefore, it is necessary to research document-level sentiment analysis. Traditional document-level sentiment analysis approaches are mainly based on sentiment lexicons and machine learning [9]. The methods based on sentiment lexicons mainly combine the sentiment inclination and intensity of words with corresponding rules to analyze the sentiment tendency of the text. However, these approaches require considerable manpower, and sentiment lexicons are unable to cover every area [10]. Machine learning methods mainly perform feature selection and train data to construct a classification model. However, these methods can be influenced by the quality of the annotated corpus and involve considerable labor costs.

With the rapid development of deep learning, methods based on neural networks have been widely employed in NLP. Compared with traditional machine learning methods, deep learning approaches do not require the construction of sentiment lexicons or manual feature engineering. These approaches automatically extract deep abstract features from text. Moreover, they have obvious advantages in terms of the construction of a classification model and effect optimization.

The recurrent neural network (RNN) [11] and convolutional neural network (CNN) [12] are current mainstream neural network methods in text sentiment analysis tasks. RNN has important applications in text sentiment analysis as it contains the temporal relation of input information. RNN can flexibly explore long input sequences and obtain text representations. Moreover, RNN can convert a text into vectors composed of feature vectors and sequence dimensions. LSTM [13] is a development of RNN with a memory unit and gate mechanism that solves the problem of long short-term dependencies and addresses the issue of gradient disappearance and gradient explosions in RNN [14]. CNN [12] applies 1D convolution to perform feature mapping and 1D pooling operation to obtain a fixed-length output, which can effectively perform text classification and enable the extraction of features between two adjacent words. Xu, J. [14] proposed cached LSTM neural networks to conduct document-level sentiment analysis. Tang [15] proposed the LSTM-GRNN model to store long-distance document information in sentiment analysis, and Rao et al. [16] proposed the SR-LSTM model with two hidden layers that removes sentences with less sentiment polarity in document-level sentiment analysis. However, none of these models are able to capture the intrinsic relations between two sentences or distinguish the importance of different sentences.

Although these deep learning models have achieved impressive results, there is still ample room for improvement. First, the 1DCNN is unable to capture long distance dependencies, so it may disregard many context relations between two sentence features. Although CNN effectively perform text classification, the task of sentiment analysis is not a simple issue of text classification because of the author's emotion reflected in the text. Second, LSTM can only scan sequences in one direction and is unable to synchronously access both past information and future information, hence, it may lack some long-term dependencies of sentences. Third, each word in a sentence, and each sentence in a document has different sentiment semantic contributions. The sentiment polarity of a document is often determined by certain key sentences, and the sentiment polarity of a sentence is usually determined by certain sentiment polarity words, such as positive and negative words [17]. Therefore, we should not consider all parts of a document to be equally important.

To address the above problems, this paper proposes a novel model named AttBiLSTM-2DCNN, which is a combination of a two-layer bidirectional long short-term memory network and a two-dimensional convolutional neural network to address document-level sentiment analysis with Attention mechanisms. First, the first bidirectional long short-term memory (BiLSTM) layer learns sentence sentiment representations from word embedding in both the forward direction and backwards direction of a sequence. The second BiL-

STM layer is used to obtain a document representation that has time-step and feature dimensions from these sentence sentiment representations. During this step, word-level and sentence-level attention mechanisms are utilized to distinguish the different contributions of words and sentences in the document. To capture more sentiment features, the model utilizes a two-dimensional convolutional neural network (2DCNN), which performs two-dimensional convolution and two-dimensional mean pooling operations. High-level document sentiment representations are obtained using the 2DCNN and are employed for document sentiment analysis.

The contributions of this study include the following three aspects:

1. This study proposes a combined framework in which the two-layer bidirectional LSTM captures long-term dependencies and sentiment semantic information in words and sentences from forward and backwards directions with two hidden layers. Moreover, the 2DCNN can extract more local contextual features to obtain high-level document sentiment representations.
2. This paper introduces word-level and sentence-level attention mechanisms. Compared with no attention mechanisms or just a one-layer attention mechanism, these mechanisms can enhance the importance of sentiment polarity words and focus on important sentences to promote the performance of text sentiment analysis.
3. The experimental results show that the model achieved better performance than certain state-of-the-art models on two document-level public review datasets (Yelp2015 and IMDB).

Our study is structured as follows: Section 2 retrospectively reviews the previous studies of sentiment classification. Section 3 focuses on the model architecture. Section 4 describes the experiments and analyses the empirical results. Conclusions are summarized in Section 5.

2. Related Work

It is widely appreciated that sentiment classification is an essential research topic in NLP and has made great achievements. Machine learning methods have been utilized in a number of previous studies [18–20]. However, with the development of the neural network method, neural networks have been used in sentiment classification by many researchers [12–15] because they achieve higher accuracy than traditional methods (e.g., machine learning). Therefore, the neural network method is selected as the experimental method in this paper. Details regarding neural networks and traditional machine learning methods are discussed in this section.

2.1. Traditional Methods

Lexicon-based methods use sentiment dictionaries and a series of relevant linguistic rules to label the sentiment polarity of words. They obtain the sentiment tendency of the document according to the total sentiment polarity [21].

Machine learning-based methods rely on classifiers that are trained by labelled text. Pang [18] et al. obtained film comment sentiment classification by a machine learning-based method and used the SVM, NB, and maximum entropy (ME) to show the performance of different classifiers. When using the support vector machine, the Unigram word feature and BOOL feature weight, relatively high classification accuracy was achieved. Agarwal et al. [22] conducted experiments on film and commodity comments, and the results showed that the Boolean polynomial NB classifier had higher classification accuracy and a shorter running time than the SVM classifier.

2.2. Deep Learning Methods

It is commonly acknowledged that CNN and RNN are two kinds of deep learning methods. These two methods have been applied to extract sentiment features and to obtain document representations in a large number of studies [23–30]. The details of these two methods are described as follows.

CNN-based approaches: The CNN is a development method in computer vision. With the development of NLP, CNN has been employed to extract semantic features. In addition, CNN differs from computer vision in that it consists of 1D convolution and 1D pooling layers. Yoon Kim [12] used the CNN with word2vec in sentence sentiment classification. Zhang et al. [23] suggested that a CNN based on rule optimization and critical learning can improve the accuracy of sentiment classification. Kalchbrenner [24] proposed a dynamic DCNN that utilizes a k-max pooling operation. Zhang et al. [25] were able to simultaneously process context and model the context multiple times using a multilayer CNN. Feng et al. [26] separately combined word features with parts of speech features, dependency syntax features, and position features to form three new combined features, and they inputted them into the multichannel CNN.

RNN-based approaches: The RNN is a very popular model in NLP as it can more effectively process variable length sequences due to its complex structure. LSTM is an extension of the RNN, which was first suggested by Hochreiter [13]. This model can solve the following three problems that the RNN cannot solve: long-short term dependency, gradient explosion, and disappearance. Xu et al. [14] proposed cached LSTM, which can store information in the sequence far from the current position. The accuracy of Tree-LSTM was found to be better than certain LSTM baseline methods, which was proposed by Tai et al. [27]. However, the limitations of this method are its dependence on the parse tree structure and many phrase-level annotations. Rao et al. [16] proposed an SR-LSTM model with two hidden layers which remove sentences with less sentiment polarity in document-level sentiment analysis. Li et al. [28] proposed a sentiment classification method based on LSTM with a self-attention mechanism and multichannel features.

Hybrid neural networks approaches: Some researchers combined two network architectures to perform text sentiment classification [29]. A combined model involving LSTM and CNN was proposed by Kim et al. [30], which was shown to improve the sentiment analysis performance. Rhanoui et al. [31] proposed a combination of CNN and BiLSTM models with Doc2vec embedding. Tang [15] proposed a combined model, LSTM-GRNN, for document sentiment classification. First, a layer of CNN or LSTM was utilized to learn sentence-level representations from word embedding. Second, he adopted a gated recurrent neural network (GRU) to encode semantic information from sentences to obtain document vector representation. Last, a softmax classifier was utilized to categorize the polarity of the document.

In RNN, an attention mechanism was used to achieve excellent results. Liu et al. [32] introduced a model that combines aspect classification and an attention mechanism with the premise of adding aspect information to content attention. Zhou [17] proposed the method of cross-language sentiment classification with a hierarchical attention mechanism, which distributes the attention mechanism to words and sentences. A multi-sentiment-resource enhanced attention network (MEAN) method was proposed by Lei et al. [33]. This method combines intensity words, sentiment lexicons, and negation words with the attention mechanism to perform sentiment classification. The model can more effectively classify emotions by using different emotion-related information. Bhuvaneshwari et al. [34] proposed a Bi-LSTM self-attention model, which applied an attention mechanism to capture n-gram features and sets different weights between words and sentences.

Despite the success of the approaches described above, we can conclude that these methods are proposed with a layer of LSTM or generic combination approaches of two neural networks. As a comparison, first, our method proposes bidirectional LSTM with two hidden layers. The difference is that BiLSTM can learn the dependences of the context from the forward and backwards directions. Using the structure of two-layer BiLSTM, we can extract more specific semantic information than when using standard LSTM. Second, the 1DCNN, which simply applies 1D convolution and 1D max pooling on the document matrix, may ignore the dependencies between features on feature dimensions and destroy the structure of the feature representation. Therefore, we use 2D convolution and 2D pooling operations to capture more local meaningful features. Third, we utilize the

attention mechanism at the word-level and sentence-level. Compared with the use of either no attention mechanism or a one-layer attention mechanism, it is shown that our model could distribute different weights to words and sentences and focus on important sentiment information.

3. AttBiLSTM-2DCNN Model

Details regarding the AttBiLSTM-2DCNN model are introduced in this section. The architecture of our model, which involves the following three parts, is shown in Figure 1.

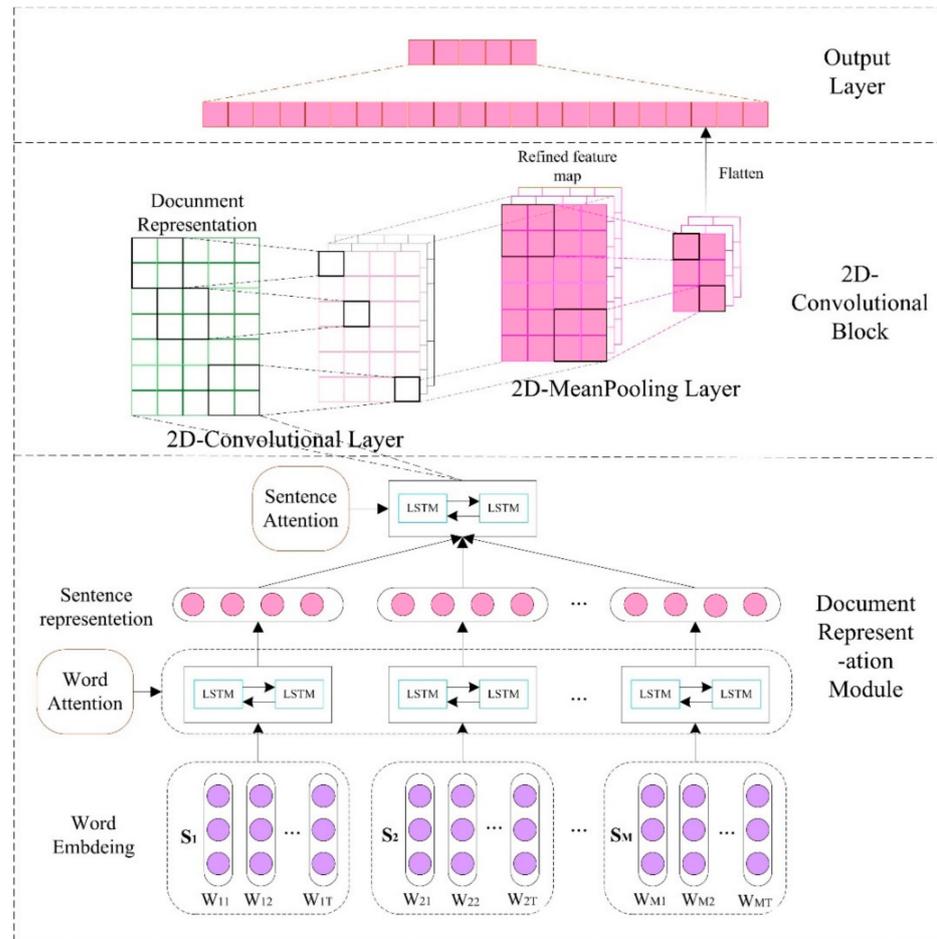


Figure 1. Structure of the AttBiLSTM-2DCNN.

Document Representation Module: We utilize bidirectional LSTM to obtain sentiment semantics. First, a layer bidirectional LSTM and a word attention mechanism are utilized to obtain sentence vector representation. Second, we obtain document matrix representation from sentence vector representation by utilizing the second layer bidirectional LSTM and a sentence attention mechanism.

Two-dimensional Convolution Block: This block contains a two-dimensional convolution operation and a two-dimensional mean pooling operation.

Output layer: A softmax classifier is utilized to divide the document sentiment representation that is obtained by this layer.

3.1. Document Representation Module

This part of the architecture is used to obtain sentence-level sentiment representation and document-level sentiment representation by using two layers of bidirectional LSTM, where the first layer obtains sentence sentiment representations based on GloVe, and the second layer acquires document-level sentiment representations from these sentence

representations. The importance of words and sentences is distinguished using a two-layer attention mechanism, which could suppress unnecessary words and sentences.

3.1.1. LSTM

Our model utilizes LSTMs to obtain the document matrix representation, which can record long-term information through memory cells and solve gradient disappearance or explosion caused by long-term dependencies.

Figure 2 shows the LSTM neural network model. i , o , c , and f are the input gate, output gate, memory cell, and forget gate, respectively. h_{t-1} is the output value at time-step $t - 1$. The formula of LSTM can be shown as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{5}$$

$$h_t = o_t * \tanh(c_t) \tag{6}$$

where σ represents the logistic sigmoid function, and i_t , f_t , o_t , c_t , and h_t are the input gate, forget gate, output gate, memory cell, and output value at time-step t , respectively. The values of the gating vectors i_t , f_t , and o_t are in $[0, 1]$. $*$ is the multiplication operation. b represents bias, and $h_t \in R_H$, $W_i, W_f \in R_{H \times d}$. $b_i, b_f, b_o, b_c \in R_H$, and $U_i, U_f, U_o, U_c \in R_{H \times H}$. H and h represent the dimensionality of the hidden and input layers.

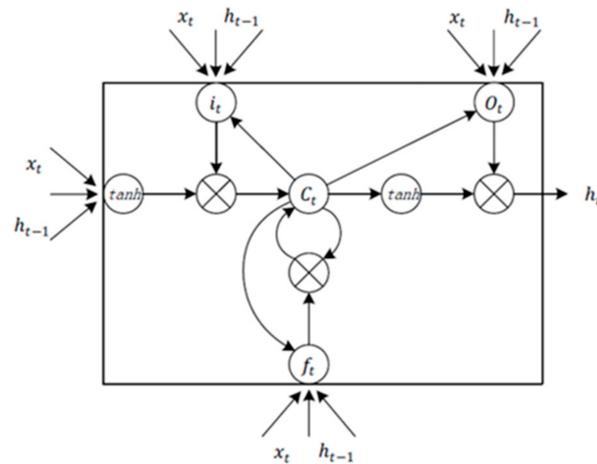


Figure 2. Structure of LSTM.

3.1.2. BiLSTM-Based Word Representation

Assume that a document is constructed of M sentences s_i , that T words create a sentence, that w_{it} is the t th words of the i th sentence, and that $t \in [1, T]$. First, each word in a sentence is embedded by the embedded matrix into a low dimensional vector representation. x_{it} is the vector representation that is the t th word in the i th sentence. Embedded learning algorithms such as Word2Vec [35], GloVe [36], or FastText [37] can pre-train an embedded matrix on the corpus. In our model, GloVe is used to obtain the grammatical relevance and semantics between two words. As BiLSTM can obtain more context information than LSTM, this paper employs a layer of bidirectional LSTM (BiLSTM) to obtain the sentence representation by exploring the semantic information between two words. BiLSTM contains two LSTMs in opposite directions. The forward LSTM studies

a sentence from the beginning to the end, while the backward LSTM is the opposite. The BiLSTM form is expressed as follows:

$$\vec{h}_{it} = \overrightarrow{\text{LSTM}}(x_{it}), t \in [1, T] \tag{7}$$

$$\overleftarrow{h}_{it} = \overleftarrow{\text{LSTM}}(x_{it}), t \in [T, 1] \tag{8}$$

The annotation and semantics of w_{it} are obtained by integrating \vec{h}_{it} with \overleftarrow{h}_{it} , which is $h_{it} = [\vec{h}_{it} \otimes \overleftarrow{h}_{it}]$.

3.1.3. Sentence Representation with Word Attention

In general, different words have different levels of importance in a sentence. Certain sentiment polarity words are decisive factors in determining the sentiment polarity of a sentence. The structure of the attention mechanism is shown in Figure 3. Words are attributed different weights in a sentence.

$$S_i = \sum_{t=1}^T \alpha_i h_i, i \in [1, M] \tag{9}$$

where α_{it} is the weight of the t th word in the i th sentence. To compute α_{it} , first, we obtain e_{it} which is a hidden representation of h_{it} :

$$e_{it} = f(W_S h_{it} + b_s) \tag{10}$$

where f is a nonlinear transformation function, W_S is a weight matrix and b_s is a bias term. Second, the weight α_{it} is computed as follows:

$$\alpha_{it} = \frac{\exp(e_{it})}{\sum_t \exp(e_{it})} \tag{11}$$

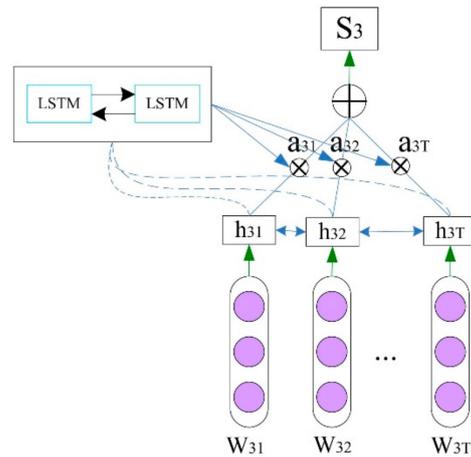


Figure 3. Structure of word-level attention. The word embedding h_{it} is sequentially put into a neural network BiLSTM, α_{it} are weights for h_{it} , and S_i is the sentence vector computed by weighting words.

3.1.4. Document Representation with Sentence Attention

A BiLSTM layer is utilized to obtain the document vector representation from the sentence vector representation, which could obtain more sentiment semantic information. The sentence representation s_i is inputted to the second BiLSTM layer to obtain document semantic representation in two directions, and the corresponding form can be expressed as follows:

$$\vec{h}_{it} = \overrightarrow{\text{LSTM}}(s_i), i \in [1, M] \tag{12}$$

$$\overleftarrow{h}_{it} = \overleftarrow{\text{LSTM}}(s_i), i \in [M, 1] \tag{13}$$

The annotation and semantics of s_i are obtained by integrating \vec{h}_{it} with \overleftarrow{h}_{it} , which is $h_i = [\vec{h}_i \oplus \overleftarrow{h}_i]$. Similarly, sentence-level attention is utilized to allocate different weights to different sentences. The document matrix representation is formed by the weighted sentence vectors and preserves richer sentiment semantic information regarding the document:

$$s_{wi} = \alpha_i h_i, i \in [1, M] \tag{14}$$

where α_i represents the weight of the i th sentence in the document. First, we calculate e_i which is a hidden representation of h_i :

$$e_i = f(W_s h_i + b_s) \tag{15}$$

where f is a nonlinear transformation function, W_s is a weight matrix, and b_s is a bias term. Second, the weight α_i of sentence s_i is computed as follows:

$$\alpha_i = \frac{\exp(e_i)}{\sum_i \exp(e_i)} \tag{16}$$

A document matrix is constructed with a weighted sentence sentiment feature vector and can be represented as follows:

$$D = [s_{w1}; s_{w2}; \dots; s_{wM}], s_{wi} \in \mathbb{R}^d \tag{17}$$

where M represents the time-step dimension and d denotes the feature dimension.

3.2. Two-Dimensional Convolution Module

To obtain the dependencies between two sentence features, our model utilizes 2DCNN in this section. As the document matrix representation can capture the sentiment semantics of a document, we consider it to capture the longer context and the compositional information of the document, similar to a 2D ‘image’, so we can utilize a two-dimensional convolution block to capture more useful information.

3.2.1. Two-Dimensional Convolution Layer

Considering that each sentence contains d feature units, document sentiment representation $D = [s_{w1}; s_{w2}; \dots; s_{wM}]$ with $D \in \mathbb{R}^{M \times d}$ is a two-dimensional ‘image’ containing $M \times d$ features. We use convolution filters $g_k \in \mathbb{R}^{k_1 \times k_2}$ to execute 2D convolution operations and each filter size has N filters. We apply g_k to capture sentiment dependencies between two sentence features. For the n th filter, feature $O_{i,j}$ is produced by the vector window $D_{i:i+k_1-1, j:j+k_2-1}$ by:

$$o_{i,j}^{n,k} = f\left(g_k^n \cdot D_{i:i+k_1-1, j:j+k_2-1} + b_i^{n,k}\right) \tag{18}$$

$i \in [1, M - K_1 + 1], j \in [1, d - K_2 + 1], \cdot$ is a dot product operation, $b_i^{n,k} \in \mathbb{R}$ is a bias term, and f is a nonlinear transfer function, so we adopt \tanh . The model applies Filter g_k to generate the feature map in matrix D :

$$O^{n,k} = \begin{bmatrix} O_{1,1}^{n,k} & \cdots & O_{1,d-k_2+1}^{n,k} \\ \vdots & \ddots & \vdots \\ O_{M-k_1+1,1}^{n,k} & \cdots & O_{M-k_1+1,d-k_2+1}^{n,k} \end{bmatrix} \tag{19}$$

where $O^{n,k} \in R^{(M-k_1+1) \times (d-k_2+1)}$. N convolution filters produce N feature maps; we use them to learn semantic features. Filter g_k can generate a three-dimensional feature map:

$$O^k = [O^{1,k}, O^{2,k}, \dots, O^{N,k}] \tag{20}$$

where $O^k \in R^{(M-k_1+1) \times (d-k_2+1) \times N}$. The feature dimension size is $(M - k_1 + 1) \times (d - k_2 + 1)$ and the size of the channel dimension is N . The size of the convolution filters could be changed to obtain more feature semantic information.

3.2.2. Two-Dimensional Pooling Operation

We use a 2D mean pooling operation to diminish and fix the feature map dimension. Given the feature map O^{n,k^n} of O^{k^n} with $O^{n,k^n} \in R^{A \times B}$, we apply a 2D mean pooling operation $p \in R^{p_1 \times p_2}$ in the windows of O^{n,k^n} to obtain the average value:

$$p_{i,j}^{n,k^n} = average(O_{i:i+p_1, j:j+p_2}^{n,k^n}) \tag{21}$$

where $average(\cdot)$ is a 2D mean-pooling function, $i \in [1, 1 + p_1, \dots, 1 + A - p_1]$ and $j \in [1, 1 + p_2, \dots, 1 + B - p_2]$. The pooling result from O^{n,k^n} is:

$$p^{n,k^n} = \begin{bmatrix} p_{1,1}^{n,k^n} & \dots & p_{1,1+B-p_2}^{n,k^n} \\ \vdots & \ddots & \vdots \\ p_{1+A-p_1,1}^{n,k^n} & \dots & p_{1+A-p_1,1+B-p_2}^{n,k^n} \end{bmatrix} \tag{22}$$

where $p^{n,k^n} \in R^{(1+A-p_1) \times (1+B-p_2)}$ and the complete feature map $p^{k^n} = [p^{1,k^n}, p^{2,k^n}, \dots, p^{N,k^n}]$ with $p^{k^n} \in R^{(1+A-p_1) \times (1+B-p_2) \times N}$.

3.3. Output Layer

The output layer obtains high-level document representation, which is a 1D vector v transformed from the 2D feature map p^{k^n} . A *softmax* classifier layer is utilized to classify the vector v and to predict the sentiment polarity y :

$$p(y|s) = softmax(W_s v + b_s) \tag{23}$$

Cross-entropy is employed as the train target, and we minimize it. Avoiding overfitting, we adopt L regularization for all parameters and L is computed as follows:

$$L = - \sum_{t \in T} \sum_{i=1}^K ground_i \cdot \log(p_i(t)) + \frac{\lambda \|\theta\|^2}{2} \tag{24}$$

where T represents the training documents, K represents the class number of target classes, i represents the class index, t is a document, and $p(t)$ represents the distribution of the projected sentiment. By using L , the aim is to minimize the cross-entropy error of $ground \in R^K$ and $p(t)$.

4. Experiments

Based on the proposed model (BiLSTM-2DCNN and AttBiLSTM-2DCNN), the results of two datasets for document-level sentiment classification and the effect of different filter window sizes are presented in this section.

4.1. Dataset and Experimental Setup

This paper selected two common datasets, namely, IMDB (<http://www.imdb.com/> (accessed on 15 January 2018)) and Yelp 2015 (http://www.yelp.com/dataset_challenge (accessed on 15 January 2018)). IMDB is a movie review dataset and Yelp 2015 is a restaurant

review dataset. Both datasets can be publicly accessed. Table 1 describes the statistical features of the above two datasets. These two datasets were classified into three parts, namely, a training set, validation set, and testing set, respectively, with 80/10/10, Sens/D, and Ws/D represent the average number of sentences and words in each document. The train size is the number of documents in the training set. The valid size is the number of documents in the validation set. The test size is the number of documents in the testing set.

Table 1. Statistical features of the IMDB and Yelp 2015 datasets.

| Dataset | Sens/D | Ws/D | Train Size | Valid Size | Test Size |
|-----------|--------|-------|------------|------------|-----------|
| IMDb | 14.02 | 325.6 | 25,001 | 2426 | 2302 |
| Yelp 2015 | 8.97 | 151.9 | 38,019 | 3725 | 4005 |

Note: Sens/D and Ws/D represent the average number of sentences and words in each document.

To realize better performance, we utilize 300-dimensional GloVe [36] as pretrained word embedding. To learn the dependent features in the 2D convolution operation and to conduct a comparative experiment, the window size of the filters is set to (3,3), (4,4), and (5,5) with 200 feature maps. Adagrad [37] is selected as an optimizer. The batch sizes of IMDB and Yelp 2015 are 128 and 64, respectively. We use a dropout operation with a dropout rate of 0.4 for the word embedding and 0.3 for the BiLSTM layer, and we also employ the L2 penalty with a coefficient of 10^{-5} over the parameters. According to the average number of sentences in IMDB and Yelp 2015 in Table 1, the maximum number of sentences of IMDB and Yelp 2015 are set to 15 and 9, respectively.

4.2. Evaluation Parameters

It is commonly appreciated that accuracy is a standard metric used to evaluate the overall sentiment analysis performance [38–42]. According to the research by Pei et al. [41], Rao et al. [16], and Behera et al. [43], this paper adds two evaluation parameters (F1-Measure and MSE) to evaluate the performance of sentiment analysis. Finally, we employ Accuracy, F1-Measure, and MSE to assess our model in this paper. There are four components that comprise different evaluation parameters.

- True Positive (TP): The number of positive labelled reviews is predicted to be positive.
 - False Positive (FP): The number of negative labelled reviews is predicted to be positive.
 - True Negative (TN): The number of negative labelled reviews is predicted to be negative.
 - False Negative (FN): The number of positive labelled reviews is predicted to be negative.
1. *Accuracy* is defined as the fraction of samples that are predicted correctly.

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

2. *Recall* is defined as the ratio of true positive predictions to the total number of actual positive samples.

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

3. *Precision* is defined as the ratio of true positive predictions to the total number of positive predictions.

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

4. *F1-Measure* is the harmonic mean of Precision and Recall.

$$F1 - Measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

5. The mean squared error (*MSE*) is a good measure of the average error. The smaller the *MSE* values are, the better the model predicts the experimental data.

$$MSE = \frac{\sum_j^N (standard_j + predicted_j)^2}{N}$$

4.3. Baseline Models

To compare the accuracy of different models, certain traditional machine learning methods and neural network methods were employed to conduct the experiment (shown in Table 2).

Table 2. Result of our model against certain competitive models on IMDB and Yelp. Accuracy (higher is better), F1-Measure (higher is better), and MSE (lower is better) are evaluation metrics.

| Model | IMDB | | | Yelp2015 | | |
|------------------------|----------|------------|------|----------|------------|------|
| | Accuracy | F1-Measure | MSE | Accuracy | F1-Measure | MSE |
| NB | 0.394 | 0.392 | 4.21 | 0.613 | 0.607 | 0.73 |
| SVM | 0.409 | 0.406 | 3.53 | 0.598 | 0.603 | 0.81 |
| CNN | 0.376 | 0.369 | 3.82 | 0.623 | 0.626 | 0.59 |
| LSTM | 0.410 | 0.412 | 3.23 | 0.617 | 0.619 | 0.67 |
| BiLSTM | 0.430 | 0.433 | 2.67 | 0.642 | 0.647 | 0.55 |
| LSTM-GRNN | 0.453 | 0.447 | 2.42 | 0.676 | 0.679 | 0.49 |
| SSR-LSTM | 0.455 | 0.450 | 2.25 | 0.678 | 0.681 | 0.48 |
| BiLSTM-2DCNN | 0.478 | 0.469 | 2.14 | 0.698 | 0.701 | 0.43 |
| AttBiLSTM-2DCNN | 0.483 | 0.486 | 2.10 | 0.705 | 0.709 | 0.40 |

NB [18] is a common traditional machine learning algorithm that uses bag-of-words features.

SVM [44] utilized bag of words as features to train the classifier.

LSTM [13] is a development of the RNN, including memory cells and three gating mechanisms.

BiLSTM [35] contains two LSTMs in opposite directions: the forward LSTM reads from the head to the end of a text, and the backwards LSTM reads from the opposite direction.

CNN [12] utilized word embedding to pretrain words and was first proposed by Kim for use in sentiment classification.

LSTM-GRNN [15] utilized a layer of CNN or LSTM to obtain the sentence representation and adopted a layer of GRNN to obtain document representation from sentence representation.

SSR-LSTM [16] utilized a layer of LSTM to obtain sentence representation and then used a layer of LSTM to obtain a document vector representation by encoding the sentence representation. When setting the number of input sentences to the maximum, the sentence representations can reserve important sentences.

4.4. Results

Table 2 presents the experimental results of the above two databases. The performance of the model is evaluated using accuracy (higher is better), F1-Measure (higher is better), and MSE (lower is better). The results of certain based methods were referenced from previous studies [13,18,35,41], and we obtained other methods' results through experiments. Based on the results in Table 2, multiple findings are obtained.

There are two methods in our paper: BiLSTM-2DCNN and AttBiLSTM-2DCNN. BiLSTM-2DCNN is a combined neural network using 2DCNN on document matrix representation, which is obtained by bidirectional LSTM. AttBiLSTM-2DCNN adds an attention

mechanism at the word and sentence levels based on BiLSTM-2DCNN. We observe that AttBiLSTM-2DCNN achieves perfect accuracy, F1-Measure, and MSE on two datasets, achieving accuracies of 48.3% and 70.5% in IMDB and Yelp2015, which means it outperforms the best baseline model, SSR-LSTM, by 2.8% and 2.7%, respectively.

Compared with NB and SVM, our two methods (BiLSTM-2DCNN and AttBiLSTM-2DCNN) have better accuracy, F1-Measure, and MSE value. The probable cause is that machine learning methods depend on the quality of the annotated corpus and labour costs. However, neural network methods can automatically learn deep features from the data, so they display better performances in sentiment classification.

Our two methods (BiLSTM-2DCNN and AttBiLSTM-2DCNN) perform better than the simple neural network methods, e.g., CNN, LSTM, and BiLSTM, and better than the combined models, e.g., LSTM-GRNN and SSR-LSTM, probably because our methods use bidirectional LSTM to capture more dependencies among the context features from the forward and backwards directions than many models that only work in a single directional. This also indicates that the two-layer BiLSTM can extract more specific semantic information than the standard one-layer BiLSTM.

In terms of whether the attention mechanisms should be used, AttBiLSTM-2DCNN is shown to achieve 48.3% and 70.5%, which outperforms BiLSTM-2DCNN by 0.5% and 0.7% on IMDB and Yelp 2015, respectively. This indicates that using an attention mechanism at the word and sentence levels can enable the network to focus on important sentiment information and ignore unimportant information, which could improve the performance of sentiment analysis in a document.

4.5. Effect of Filter Window Size

To achieve better performance, we explore the effect of the size of the 2D convolution filter window, considering sizes of 2×2 , 3×3 , 4×4 , 5×5 , 6×6 , and 7×7 . The results are shown in Figure 4, and this analysis is conducted using both datasets and uses AttBiLSTM-2DCNN with feature maps of 200.

Figure 4 shows that the two datasets have different optimal filter window sizes. The Yelp2015 dataset could achieve the best properties when the filter size is 4, and the F1-Measure decreases when the size is away from 4. This probably because if the size is less than 4, the model cannot capture enough sentiment semantic information and relevant dependencies between two sentence features. Furthermore, the model may bring too much redundant information when the size is too large. As IMDB contains more sentences than Yelp 2015 (shown in Table 1), the optimal filter window of IMDB is larger than Yelp2015, with a value of 5.

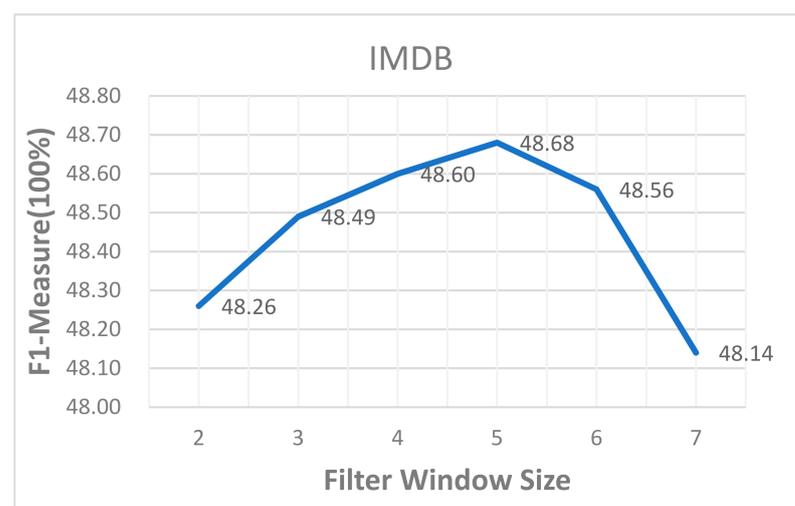


Figure 4. Cont.

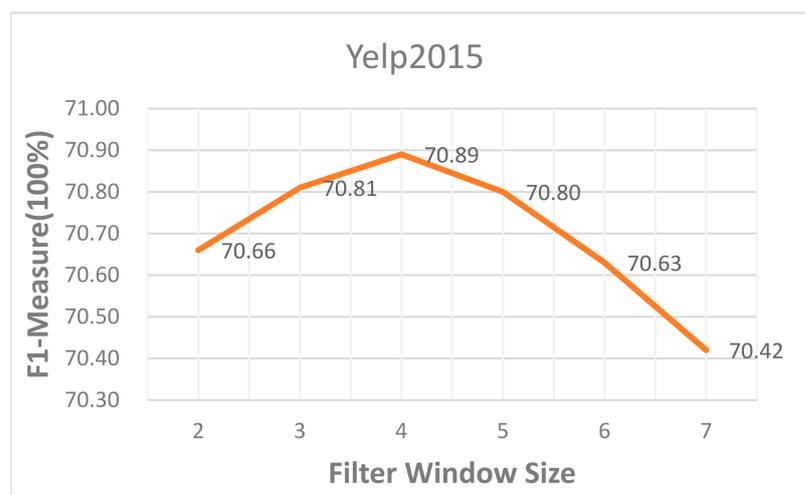


Figure 4. Effect of filter window size.

5. Conclusions

This paper proposes two new neural network models (BiLSTM-2DCNN and AttBiLSTM-2DCNN) to classify the sentiment of documents. BiLSTM-2DCNN is a model that combines two-layer, bidirectional LSTM, and a 2DCNN. First, the approach encodes the sentiment semantics and relationships in a sentence into a low-level document matrix representation. Second, we utilize the 2DCNN to investigate the dependencies of sentences in the document representation. AttBiLSTM-2DCNN adds an attention mechanism to allocate different weights to words and sentences based on their contributions. We conduct our experiment on the IMDB and Yelp 2015 datasets. The experimental results indicate that our models are efficient and outperform certain other models. This finding further proves that (1) the simple deep learning methods (CNN and LSTM) cannot improve the performance of sentiment analysis, while by combining two neural networks, more sentiment semantic information can be obtained, (2) compared with the 1DCNN, the 2DCNN can obtain a more dependent relationship between two sentences, which can increase the accuracy of the method, and (3) distributing different weights to different words and sentences can also improve the sentiment analysis performance.

Author Contributions: Conceptualization, Y.M. and Y.Z.; methodology, Y.M. and Y.Z.; validation Y.M.; formal analysis, H.Z. and L.J.; investigation, H.Z. and L.J.; data curation, Y.M.; writing—original draft preparation, Y.M.; writing—review and editing, Y.Z. and L.J.; supervision, Y.M., Y.Z. and H.Z.; project administration, Y.M., Y.Z. and L.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (Grant No. 71901043), Humanities and Social Science project of Ministry of Education of China (Grant No. 21YJC630169), Natural Science Foundation of Chongqing (Grant No. cstc2021jcyj-msxmX1010).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the research team members for their contributions to this work.

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

References

1. Chen, C.; Zhuo, R.; Ren, J. Gated recurrent neural network with sentimental relations for sentiment classification. *Inf. Sci.* **2019**, *502*, 268–278. [[CrossRef](#)]
2. Sudhir, P.; Suresh, V.D. Comparative study of various approaches, applications and classifiers for sentiment analysis. *Glob. Transit. Proc.* **2021**, *2*, 205–211. [[CrossRef](#)]
3. Minaee, S.; Azimi, E.; Abdolrashidi, A.A. Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models. *arXiv* **2019**, arXiv:1904.04206.
4. Dang, N.C.; Moreno-García, M.N.; De la Prieta, F. Sentiment analysis based on deep learning: A comparative study. *Electronics* **2020**, *9*, 483. [[CrossRef](#)]
5. Yang, L.; Li, Y.; Wang, J.; Sherratt, R.S. Sentiment analysis for E-commerce product reviews in Chinese based on sentiment lexicon and deep learning. *IEEE Access* **2020**, *8*, 23522–23530. [[CrossRef](#)]
6. Pathak, A.R.; Agarwal, B.; Pandey, M.; Rautaray, S. Application of deep learning approaches for sentiment analysis. In *Deep Learning-Based Approaches for Sentiment Analysis*; Springer: Singapore, 2020; pp. 1–31.
7. Lin, Y.; Chen, W.; Xue, L.; Zuo, W.; Yin, M. A survey of sentiment analysis in social media. *Knowl. Inf. Syst.* **2018**, *60*, 617–663.
8. Huang, F.; Li, X.; Yuan, C.; Zhang, S.; Zhang, J.; Qiao, S. Attention-emotion-enhanced convolutional LSTM for sentiment analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–14. [[CrossRef](#)]
9. Patel, R.; Passi, K. Sentiment analysis on Twitter data of world cup soccer tournament using machine learning. *IoT* **2020**, *1*, 14. [[CrossRef](#)]
10. Appel, O.; Chiclana, F.; Carter, J.; Fujita, H. Cross-ratio uninorms as an effective aggregation mechanism in sentiment analysis. *Knowl. Based Syst.* **2017**, *124*, 16–22. [[CrossRef](#)]
11. Elman, J.L. Distributed representations, simple recurrent networks, and grammatical structure. *Mach. Learn.* **1991**, *7*, 195–225. [[CrossRef](#)]
12. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
13. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
14. Xu, J.; Chen, D.; Qiu, X.; Huang, X. Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification. *arXiv* **2016**, arXiv:1610.04989.
15. Tang, D.; Bing, Q.; Liu, T. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015.
16. Rao, G.; Huang, W.; Feng, Z.; Cong, Q. LSTM with sentence representations for document-level sentiment classification. *Neurocomputing* **2018**, *308*, 49–57. [[CrossRef](#)]
17. Zhou, X.; Wan, X.; Xiao, J. Attention-based lstm network for cross-lingual sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 247–256.
18. Pang, B.; Lee, L. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.* **2008**, *2*, 1–135. [[CrossRef](#)]
19. Turney, P.D. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *arXiv* **2002**, arXiv:cs/0212032.
20. Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs up? Sentiment Classification Using Machine Learning Techniques. *arXiv* **2002**, arXiv:cs/0205070.
21. Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K.D.; Stede, M. Lexicon-Based Methods for Sentiment Analysis. *Comput. Linguis* **2011**, *37*, 267–307. [[CrossRef](#)]
22. Agarwal, B.; Mittal, N. Optimal Feature Selection for Sentiment Analysis. In *International Conference on Intelligent Text Processing & Computational Linguistics*; Springer: Berlin/Heidelberg, Germany, 2013.
23. Zhang, B.; Xu, X.; Li, X.; Chen, X.; Ye, Y.; Wang, Z. Sentiment analysis through critic learning for optimizing convolutional neural networks with rules. *Neurocomputing* **2019**, *356*, 21–30. [[CrossRef](#)]
24. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. *arXiv* **2014**, arXiv:1404.2188.
25. Zhang, S.; Xu, X.; Pang, Y.; Han, J. Multi-layer attention based CNN for target-dependent sentiment classification. *Neural Process. Lett.* **2020**, *51*, 2089–2103. [[CrossRef](#)]
26. Feng, Y.; Cheng, Y. Short text sentiment analysis based on multi-channel CNN with multi-head attention mechanism. *IEEE Access* **2021**, *9*, 19854–19863. [[CrossRef](#)]
27. Tai, K.S.; Socher, R.; Manning, C.D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. *arXiv* **2015**, arXiv:1503.00075.
28. Li, W.; Qi, F.; Tang, M.; Yu, Z. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing* **2020**, *387*, 63–77. [[CrossRef](#)]
29. Liu, F.; Zheng, J.; Zheng, L.; Chen, C. Combining attention-based bidirectional gated recurrent neural network and two-dimensional convolutional neural network for document-level sentiment classification. *Neurocomputing* **2020**, *371*, 39–50. [[CrossRef](#)]
30. Kim, Y.; Jernite, Y.; Sontag, D.; Rush, A.M. Character-Aware Neural Language Models. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.

31. Rhanoui, M.; Mikram, M.; Yousfi, S.; Barzali, S. A CNN-BiLSTM model for document-level sentiment analysis. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 48. [[CrossRef](#)]
32. Liu, Q.; Zhang, H.; Zeng, Y.; Huang, Z.; Wu, Z. Content Attention Model for Aspect Based Sentiment Analysis. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018.
33. Lei, Z.; Yang, Y.; Yang, M.; Liu, Y. A Multi-sentiment-resource Enhanced Attention Network for Sentiment Classification. *arXiv* **2018**, arXiv:1807.04990.
34. Bhuvaneshwari, P.; Rao, A.N.; Robinson, Y.H.; Thippeswamy, M.N. Sentiment analysis for user reviews using Bi-LSTM self-attention based CNN model. *Multimed. Tools Appl.* **2022**, *81*, 12405–12419. [[CrossRef](#)]
35. Mikolov, T. Distributed Representations of Words and Phrases and their Compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3111–3119.
36. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qata, 25–29 October 2014.
37. Duchi, J.; Hazan, E.; Singer, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *J. Mach. Learn. Res.* **2011**, *12*, 257–269.
38. Jurafsky, D.S.; Martin, J.H. *Speech and Language Processing*; Prentice Hall: Hoboken, NJ, USA, 2010.
39. Manning, C.D. Foundations of statistical natural language processing-table of contents. *Nat. Lang. Eng.* **2002**, *26*, 91–92.
40. AlBadani, B.; Shi, R.; Dong, J.; Al-Sabri, R.; Moctard, O.B. Transformer-Based Graph Convolutional Network for Sentiment Analysis. *Appl. Sci.* **2022**, *12*, 1316. [[CrossRef](#)]
41. Pei, Y.; Chen, S.; Ke, Z.; Silamu, W.; Guo, Q. AB-LaBSE: Uyghur Sentiment Analysis via the Pre-Training Model with BiLSTM. *Appl. Sci.* **2022**, *12*, 1182. [[CrossRef](#)]
42. Ain, Q.T.; Ali, M.; Riaz, A.; Noureen, A.; Kamran, M.; Hayat, B.; Rehman, A. Sentiment analysis using deep learning techniques: A review. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 424–433.
43. Behera, R.K.; Jena, M.; Rath, S.K.; Misra, S. Co-LSTM: Convolutional LSTM model for sentiment analysis in social big data. *Inf. Process. Manag.* **2021**, *58*, 102435. [[CrossRef](#)]
44. Joachims, T. Transductive inference for text classification using support vector machines. *ICML* **1999**, *99*, 200–209.