Hindawi
Scientific Programming
Volume 2023, Article ID 9783528, 1 page
https://doi.org/10.1155/2023/9783528

Retraction

Retracted: Emotion Analysis of Literary Works Based on Attentional Mechanisms and the Fusion of Two-Channel Features

Scientific Programming

Received 18 July 2023; Accepted 18 July 2023; Published 19 July 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Y. Han, "Emotion Analysis of Literary Works Based on Attentional Mechanisms and the Fusion of Two-Channel Features," *Scientific Programming*, vol. 2022, Article ID 8237466, 9 pages, 2022.

Hindawi
Scientific Programming
Volume 2022, Article ID 8237466, 9 pages
https://doi.org/10.1155/2022/8237466

Research Article

Emotion Analysis of Literary Works Based on Attentional Mechanisms and the Fusion of Two-Channel Features

Yanli Han [1,2]

[1] Henan University of Urban Construction, Pingdingshan, Henan 467000, China
[2] University Sains Malaysia, Penang 11800, Malaysia

Correspondence should be addressed to Yanli Han; 30140617@hncj.edu.cn

Received 25 July 2022; Revised 22 August 2022; Accepted 29 August 2022; Published 13 September 2022

Academic Editor: Lianhui Li

In order to analyze sentiment data of foreign literary works, this paper proposes an algorithm for sentiment classification of literary works. We do this by fusing different features of literary works, which in turn captures more feature information for the classifier. As traditional word embedding models are difficult to achieve fusion with the sentiment features of literary works, we consider a multifeature fusion approach of word embedding features and lexical features of literary works. A two-channel and single-channel comparison is also used to analyze the classification accuracy based on the two feature fusion methods, and a parallel CNN and BiLSTM-attention two-channel neural network model proposed. Finally, the proposed model was evaluated using a real dataset of sentiment reviews of literary works and compared with different classification algorithms in the experiments. The experimental results show that the new hybrid approach has better classification accuracy, recall, and F1 metrics. The proposed methodology is an important guide for the creation of literary works and their screenplays, as it can be used to judge whether a work appeals to readers and, importantly, can also be considered as one of the criteria for the success of a film adaptation of a literary work.

1. Introduction

Since the beginning of the new era, people's attention to contemporary literature has gradually increased, both in terms of genre and content, which is very different from traditional literature, especially in the three areas of values, ideology, and emotional values. Modern and contemporary foreign literature has a more humanistic approach to emotional values than traditional literature, as well as reflects the individual emotions of the author. In the development of foreign literature, the embodiment of emotional value in modern and contemporary foreign literature is more focused on the experience, and through reading it, one can understand the humanistic and personal feelings embedded in the work and thus grasp the author's personality and spiritual experience. In contrast to traditional foreign works, contemporary foreign literature is more open in its expression and more emotive.

A popular foreign work of literature contains a wealth of emotion. The extraction of emotion from a literary work has become a reality, and it is vital to systematically analyze the time series of emotion extracted [1]. Emotion is the soul of a literary work, and existing studies have largely addressed the importance of emotion to a literary work from a qualitative perspective and how the emotion of a literary work changes. This paper will use complexity science methods to quantitatively analyze emotion in foreign literature and its applications.

Online reviews of e-commerce, such as consumer-initiated comments on the quality of goods and services, etc., play a role in the purchasing decisions of potential consumers and directly influence the user stickiness of e-commerce platforms. The process of mining these reviews for positive and negative attitudes in order to identify people's propensity to buy a product is known as sentiment analysis [2] (SA). The goal of sentiment analysis is to extract all the points of view from a document that contains information about the points of view, an analysis method that uses natural language processing and text mining techniques, a method that takes a large amount of data and analyses it to get an understanding of something. In the early days, sentiment analysis was solved by constructing classifiers methods. Leeuw et al. [3] first proposed to solve the problem of sentiment classification using machine learning algorithms including parsimonious Bayes (NB), they used n-grams models and lexical properties to extract features of film reviews [4]. The main contribution of the fine-grained sentiment analysis method is to extract the corresponding features by syntactic analysis and to compare the experiments with the TF-IDF benchmark model, and their proposed model improves the accuracy (precision), recall (recall), and F1-score in positive or negative evaluations. Wei et al. [4, 5] extracted the features of Chinese hotel reviews by word embedding and put them into classifiers plain Bayesian (NB), support vector machine (SVM), and CNN for comparison, where SVM performed the best in classification. The word embedding approach can extract key information and hierarchical information of words from the comment text, but it cannot extract the information of emotions expressed in the words, so fusing the two features can express the information in the comment more comprehensively.

Now many scholars are very enthusiastic about applying deep learning models to sentiment analysis, applying ever-improving classical deep learning models to this domain, and even proposing new deep learning models in order to solve problems in this domain. TextCNN models were proposed earlier by Mosleh [6], using CNN algorithm to deal with the problem of sentence classification, and obtained better results than previous studies in four out of seven tasks. Liu et al. [7] used Glove to extract features and put them into a very deep CNN model for Twitter sentiment classification experiments, and the results showed that their model had higher accuracy and F1 values than the baseline model. In addition to CNNs, the long short term memory (LSTM) algorithm in deep learning, which is believed to be better at learning contextual information from text, has also been applied to the problem of sentiment classification, and Mansaray et al. [8] used LSTMs to replace the pooling layer in CNNs to perform binary and quintuple classification experiments with higher accuracy than the previously proposed models. In addition, many scholars have combined multiple features in text [9] to achieve better classification results. Nguyen et al. [10] fused word embedding features, sentiment information features, and linguistic knowledge features and overcame the disadvantages of word embedding based on relevant strategies, and their model has advantages over other classical methods. The feature fusion method used in this paper is implemented using the Syuzhet R language package. This is done by extracting the sentiment time series of a literary work to obtain the sentiment score of each sentence in the literary work and then using channel fusion and attention mechanisms to quantitatively analyze the sentiment of a variety of foreign literary works.

2. Model Principles

Since the word embedding model cannot contain the sentiment information of words well, this paper proposes the PWCNN model and PW2CNN model to apply the combination of lexical features to make the information contained in the features richer. While a single CNN model cannot capture the temporal information of a sentence well, adds an attention mechanism, and proposes a parallel classification algorithm PW2CNN and BiLSTMatt model.

2.1. *Word2vec + CNN Model.*

The classical model of TextCNN was [11] proposed by Phan, where each word is a one dimensional vector and a sentence forms a matrix. Convolutional operations are performed by different convolutional kernels, then dimensionality reduction is performed by pooling operations, and finally, binary classification is performed using a sigmoid function.

Earlier, the bag of words (BOW) model represented a vector of word frequencies for each sentence by making a bag of all the words in the document and constructing a corresponding dictionary so that the resulting vector corresponds to the dictionary and represents the frequency of the words. In longer sentence sets, this approach results in a very sparse matrix and the loss of information about the order of words. The word embedding model word2vec takes better account of word position relations [12–14] and solves the problem of oversparsity when vectorizing words with one-hot encoding.

The word embedding model is proposed to better learn word representations quickly and accurately on a large dataset, using the idea of embedding high-dimensional word vectors into a low-dimensional space so that words with adjacent meanings have closer spatial distances. By improving on the BOW model, the continuous bag-of-words model (CBOW) produces a matrix with a dimension of 256×128, which is initially set in the article, with each matrix text represented by 128 dimensions so that the concatenation of words can form a feature vector matrix, which is filled with zeros in the case of a sentence with no more than 256 words.

The word2vec + CNN model is a modification of the classical TextCNN model, where the CNN structure is convolved on the feature matrix of each sentence by a number of different 4×4 convolution kernels. The input text matrix $X \in R^{N\times N}$and filter $W \in R^{N\times V}$, and the two-dimensional convolution is given as follows:

$$c_{ij} = \sum_{U}^{u=0} \sum_{V}^{v=0} w_{uv} x_{i-u+1,j-v+1}. \tag{1}$$

Equation (1) represents the process of multiplying the elements of the text matrix $Xx_{ij}$and the elements w_{uv} of the convolution kernel matrix W to obtain the features c_{ij} using a filter W with a window size of $U \times V x_{i,j}$ to, $x_{i-u+1,j-v+1}$, which results in multiple feature matrices.

The first layer in Figure 1 is the input layer, the last layer is the output layer, and the layers in between represent the structure [15] of the CNN. This model has two convolutional layers and one pooling layer. The first has 32 convolutional kernels, all of which are 4×4 matrices, and the activation function is a ReLu function. The second convolutional layer has 16 convolutional kernels, all of which are 4×4 in size,

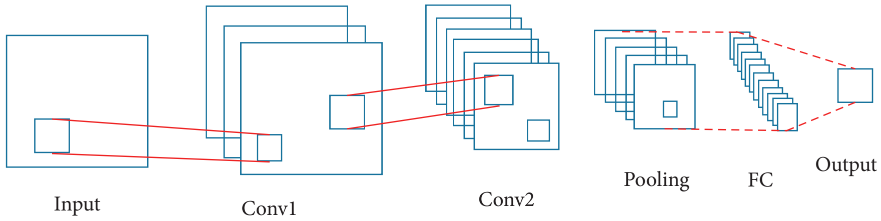

Figure 1: CNN model in 1word2vec + CNN model.

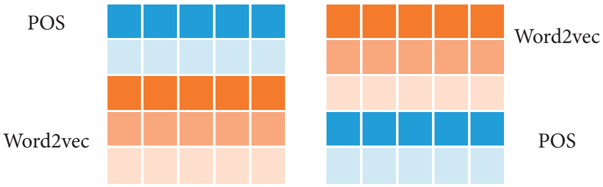

Figure 2: Word vector and lexical vector splicing diagram for word embedding model training.

and also uses the ReLu function. The reason for using the ReLu function is that it is less computationally intensive than the other activation functions and provides better protection against overfitting. The third layer is the maximum pooling layer, with a pooling kernel size of 2 × 2.

2.2. PWCNN Model. In this experiment, the word2vec + CNN benchmark model is a single-channel CNN model with word2vec as input, and has the same structure as the CNN model in the two-channel PWCNN model, except for the input of the features [16, 17]. The two features are the word embedding word2vec model matrix (256 × 128 × 1 dimension) and the part-of-speech (POS) feature input matrix (220 × 56 × 1 dimension).

The lexical annotation determines the lexical annotation of each word in the sentence and uses lexical features to disambiguate on the basis of participles. These lexical words are better able to express the subjective feelings of the reviewer, so the model focuses more on adverbs, verbs, and adjectives. The more important lexical features that the model focuses on are adjectives, adverbs, and verbs, as these lexical features are better at conveying the subjective feelings of the commenter. In addition, punctuation is also important in a sentence and thus is not considered for removal of punctuation when performing subordination and giving the lexicality of the corresponding subword. The PWCNN model splices the word vectors trained by the word embedding word2vec model with the lexical vectors, including two splicing methods, as shown in Figure 2 and Equation (2) [18].

$$X = [X_W, X_P], \tag{2}$$

where X_W denotes the word embedding word2vec model matrix; X_P denotes the lexical annotation feature input matrix.

In addition, the POS features and word2vec features of each sentence are stitched together one by one so that the whole feature matrix is meaningful. The spliced feature vectors are then trained in the same single-channel CNN model as the word2vec + CNN model, mainly to directly compare [19] the experimental results of the feature fusion single-channel model with the benchmark model. Given the difference in structure between the CNN model and the classical TextCNN model, this “top-down” rather than “left-right” splicing was chosen, but the “left-right” splicing is also worth trying. The stitching is also worth trying. At the same time, it was found that the two splicing methods have little effect on the experimental results and made no difference to the CNN. The word vector of fused features was put into the same structure of convolution as mentioned above for the experiments, and the activation function and parameters were set in the same way. The information in the feature vector is extracted through a two-layer convolutional layer, then further reduced and aggregated through a pooling layer, where full connectivity is performed through a random dropout, and to further prevent overfitting (reducing parameters), a dropout is added after this flatten layer, and finally, classification is performed. The experimental model is shown in Figure 3.

This experiment is compared with the word2vec + CNN model to clearly determine whether POS features can play a role in this classification experiment. The experiments prove that the input matrix with POS features does have better classification results. The next step is to further consider a two-channel CNN model based on this single-channel model and explore whether the same features are affected when they enter the same structural CNN model separately.

2.3. PW2CNN Model. It is well known that the two-channel model differs from the single-channel model in terms of the input method. In the single-channel model, the input is spliced with multiple features to obtain the fused features, while in the two-channel model, it is obtained by inputting different features separately [20, 21], and feature fusion can be achieved by matrix splicing. The two-channel CNN model has a matrix of POS features on one side and a matrix of word2vec features on the other side. The two feature matrices are processed separately by the CNN (two convolutional layers, a pooling layer), and the two processed features are “flattened” in the transition layer (the process of turning multidimensional data into one dimensional) and “left-right.” The purpose of this is also to preserve the information on both sides.

$$C = [C_1, C_2], \tag{3}$$

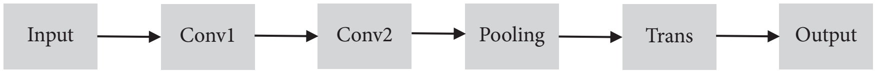

Figure 3: PWCNN model.

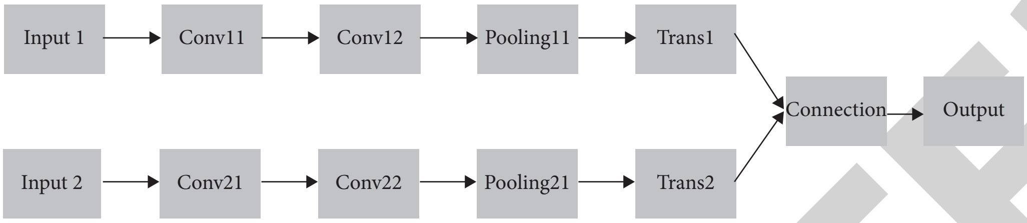

Figure 4: PW2CNN model.

where C_1 denotes the convolutional feature matrix with POS features as input; C_2 denotes the matrix convolutional feature matrix with word2vec model as input.

The overall structure of the model is shown in Figure 4.

In order to ensure comparability between the models, the structure of the CNNs for the dual and single channels was basically kept the same [22], and the comparison of the experimental results revealed that the improvement in the correctness and F1 values on their validation sets was not significant. However, when the CNN on the dual channel side of the model was replaced with a BiLSTM, the results changed significantly, indicating that it is not only the selection of features (the selection of information) but also the selection of the classifier that affects the overall results. When one side of the input is a POS feature, the CNN is able to extract the local features very well. When the other side of the input is a word2vec model, a BiLSTM model that takes into account the context is used, as the word embedding features themselves have information about the word itself. In summary, such a classifier has better experimental results and is the reason why this model is proposed in this paper.

2.4. PW2CNN and BiLSTMatt Models. The proposed PW2CNN and BiLSTMatt model uses the word2vec model [23] for word embedding, CNN, and BiLSTM models for classifier, and incorporates an attention mechanism. As the RNN (recurrent neural network) will lose its ability to learn long-term information, i.e., there is a long-term dependency problem, as the gradient will disappear or explode after multiple propagations with increasing input time series. For this reason, LSTM [24], is introduced into the model design. LSTM removes or adds information to the cell states through a gate structure.

A bidirectional long and short-term memory network (BiLSTM) usually consists of two LSTMs connected, including a positive and a negative one. The positive LSTM captures the past information in the sentence and the negative LSTM acquires the future information in the sentence. In this way, the model is able to extract contextual information, and therefore, the prediction results of the bidirectional LSTM will be more accurate. For the problem of sentiment classification of text, the model is very suitable, as it contains all the forward and backward information in the sentence.

$$
\begin{aligned}
i_t &= \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) & c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
f_t &= \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right) & o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right). \\
\tilde{c}_t &= \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) & h_t &= o_t \odot \tanh C_t
\end{aligned}
\tag{4}
$$

The LSTM network introduces a new internal state $(internalstate)c_t \in R^D$ for information transfer and outputs information to the state $h_t \in R^D$ of the hidden layer b_i, where $f_t \in [0,1]^D$, $i_t \in [0,1]^D$, and $O_t \in [0,1]^D$ are three gates to control the path of information transfer and $\odot$ is the product of vector elements. c_{t-1} is the product of vector elements. $\tilde{c}_t \in R^D$ is the memory unit of the previous moment.

The attention mechanism (AM) [25, 26] is inspired by the human cognitive function of extracting and receiving small portions of important information from a large amount of information and ignoring the rest. Similarly, the essence of the AM is to focus on certain key parts of the input and give them a higher weight. The hidden state of an LSTM model based on the AM at any given moment depends not only on the state of the hidden layer at the current moment and the output at the previous moment but also on the contextual features, which are obtained by a weighted average. This is calculated as follows:

(a) Calculate the attention distribution.

$$
a_i = p(z = i \mid \mathbf{X}, \mathbf{q}) = softmax\left(s\left(x_i, q\right)\right) = \frac{\exp\left(s\left(x_i, q\right)\right)}{\sum_{N}^{j=1} \exp\left(s\left(x_i, q\right)\right)}. \tag{5}
$$

(b) A weighted average of the input information is calculated based on the attention distribution.

$$
att(\mathbf{X}, \mathbf{q}) = \sum_{N}^{i=1} \alpha_i x_i = E_{z\sim p(z|X,q)}[x]. \tag{6}
$$

To calculate the attention distribution is to calculate the probability of selecting the first input vector given the query

Table 1: Information on crawled literature data.

Category	Literary works	Film and television works	Proportion (literature/Film)
Don quixote	4282	1218	3.52
Childhood	4002	2592	1.54
On Earth	4624	1124	4.11
War and peace	5007	1949	2.60
Red and black	4000	2332	1.72

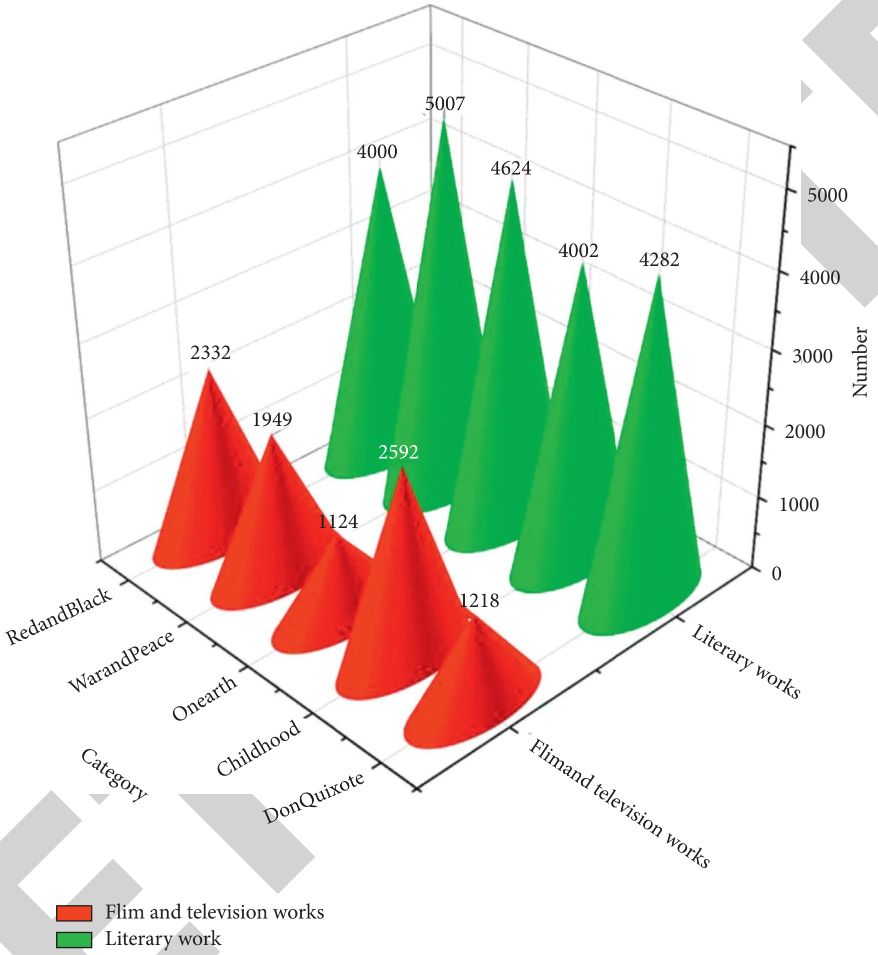

Figure 5: Comparison of the emotional evaluation of literary works and film works.

vector $a_i\mathbf{q}$ and the input $\mathbf{X}$. Where: $z \in [1, N]$ is the attention variable. The first $z = ii$ input vector is selected; a_i is the AM; $s(x_i, q)$is the attention scoring function.

The CNN and BiLSTM were chosen as parallel structures for the model classifier. The CNN model was able to extract a certain class of lexical features (e.g. adjectives, adverbs, nouns) that are more significant for the expression of emotions. The BiLSTM is more suitable for capturing temporal information features [27].

The current state in the bidirectional LSTM should also be related to the contextual features, and the output is influenced by the weighted average of the hidden states of all moments fed into the current state through the attention mechanism. The weights in the attention mechanism are adjusted according to the difference between the output and the real situation. In fact, it can be demonstrated through this experiment that this parallel model gives the best classification results compared to the benchmark model, which not only makes full use of the different feature information but also takes advantage of the different neural network models [28, 29]. This model gives better classification results than the two-channel CNN model, suggesting that the BiLSTM model with the attention mechanism plays an important role.

3. Experimental Preparation

3.1. Data Sets. The datasets used for the experiment were sentiment data from five literary works: Don Giovanni, Boyhood, On Earth, War and Peace, and Red and Black. Table 1 gives information on the number of people, who received positive reviews before and after the five works, were adapted for film and television. The sentiment of each

Table 2: Parameter setting of CNN.

Convolutional layer	Number of convolution kernels	Convolution kernel size	Activation functions
Conv1	32	4 ∗ 4	ReLu function
Conv2	16	4 ∗ 4	ReLu function
Pooling layer	Pooling methods	Pooling kernel size	Dropout function
Pool1	Maximum pooling	2 ∗ 2	0.25

Table 3: F1 values for different experimental models.

Type of experiment	F1 value (%)			
	BASELINE model	PWCNN model	PW2CNN model	PW2CNN & BiLSTMatt model
1	0.942	0.946	0.957	0.960
2	0.944	0.944	0.959	0.963
3	0.948	0.945	0.952	0.954
4	0.945	0.952	0.949	0.961
5	0.947	0.946	0.960	0.963
6	0.940	0.947	0.940	0.963
7	0.946	0.951	0.943	0.950
8	0.942	0.953	0.945	0.955
9	0.944	0.952	0.943	0.957
10	0.945	0.950	0.944	0.959

Table 4: Accuracy of different experimental models.

Type of experiment	Accuracy rate (%)			
	BASELINE model	PWCNN model	PW2CNN model	PW2CNN & BiLSTMatt model
1	0.942	0.944	0.950	0.964
2	0.944	0.944	0.951	0.963
3	0.946	0.941	0.952	0.962
4	0.943	0.952	0.945	0.961
5	0.943	0.942	0.956	0.963
6	0.940	0.947	0.951	0.962
7	0.946	0.951	0.949	0.960
8	0.942	0.953	0.945	0.957
9	0.944	0.948	0.952	0.965
10	0.945	0.950	0.946	0.960

film line was extracted using Syuzhet [30], and the sentiment indices of the novel and film lines were compared to correlate the Hurst indices of the film lines with the Rotten Tomatoes and IMDB ratings.

The results of the comparison of the number of literary works and film and television works that received positive reviews are shown in Figure 5.

3.2. Experimental Environment. In order to validate the performance metrics of the model, the comparison environment was set up as follows: Windows 64 operating system, 64 GB of memory, Intel(R) Xeon(R) CPU E5-2 650v4 @ 2.20 GHz (2 processors), and the model used the Keras deep learning framework.

3.3. Loss Function. The loss function used in the experiments in this paper is a cross-entropy loss function (binary cross-entropy), with the following equation.

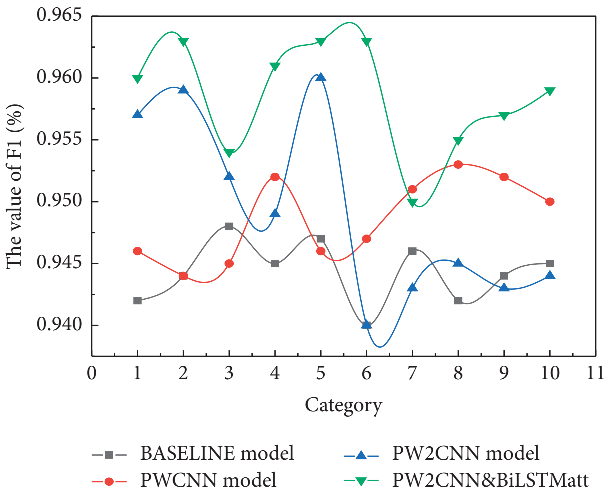

Figure 6: Comparison of the different F1 values for the four models.

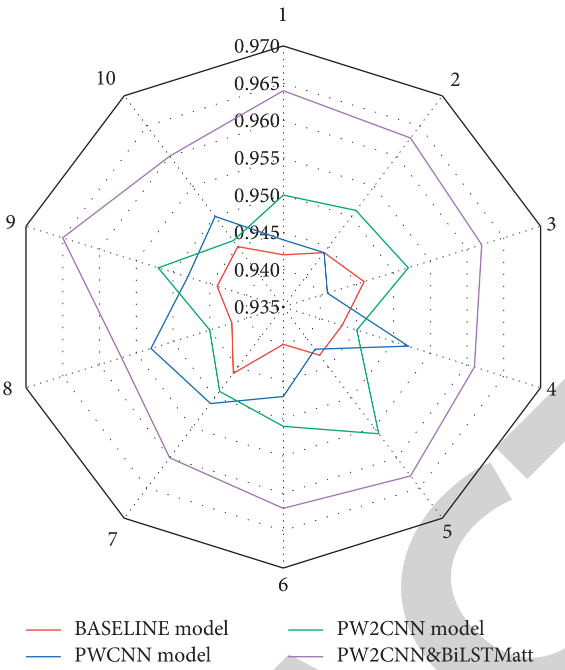

Figure 7: Comparison of the different accuracy rates of the four models.

Table 5: t-test among four models.

Models	BASELINE	PWCNN	PW2CNN	PW2CNN and BiLSTMatt
PWCNN	0.1339	0.3458	0.3141	0.4123
PW2CNN	0.2128	0.36937	0.2821	0.3896
PW2CNN and BiLSTMatt	0.28009	0.3596	0.2431	0.4012

$$loss = -\sum_{n}^{i=1} y_i \lg \widehat{y}_i + (1 - y_i)\lg(1 - \widehat{y}_i). \qquad (7)$$

where: y_i is the true discrete category; $\widehat{y}_i$ is the conditional probability distribution of the predicted category labels.

3.4. Experimental Parameters. The hyperparameters on the CNN model in this paper are set as shown in Table 2.

The PW2CNN and BiLSTMatt model puts the long memory model into a parallel structure and adds the attention mechanism, the activation function (AF) of the attention layer in the parallel structure is the softmax function, and the AF of the middle output layer is the ReLu function, and the dropout value of the attention layer is 0.3. For the model with a nonparallel structure, the FC layer and the output layer are also followed by the FC layer and the sigmoid function, respectively, with the dropout value set to 0.5 for the FC layer. The output layer is also followed by the FC layer and the output layer with the same parameters as above, mainly to maintain consistency across multiple models. Four different comparison experiments showed that the best results were obtained with the cross-entropy function as the loss function. In this case, the batch size is set to 64, and the model converges when the epoch is 10, so the epoch is set to 10.

4. Experimental Results and Analysis

4.1. Multimodel Testing. The results of 10 experiments with the four models were compared by the method mentioned previously, and the F1 values and accurate information of the four models are given in Tables 3 and 4. The BASELINE benchmark model is a word2vec + CNN model, the PWCNN is a CNN model with feature fusion, the PW2CNN model is a two-channel CNN model, and the PW2CNN and BiLSTMatt model is a parallel CNN and BiLSTM model (with attention added).

A visual comparison of the F1 values of the four models is shown in Figure 6, and information on the accuracy data of the different experimental models is given in Table 4.

A visual comparison of the accuracy of four of these models is shown in Figure 7.

4.2. Significance Tests between Different Models. The t-test is a method of significance testing that uses a small probability counterfactual method of logical reasoning to determine whether the hypothesis is valid. This test can be used to test the degree of difference between two sample means. The reason for using *the t*-test here is to test whether there is a difference between the experimental results of the different models. Table 5 shows the *t*-test values between the four models.

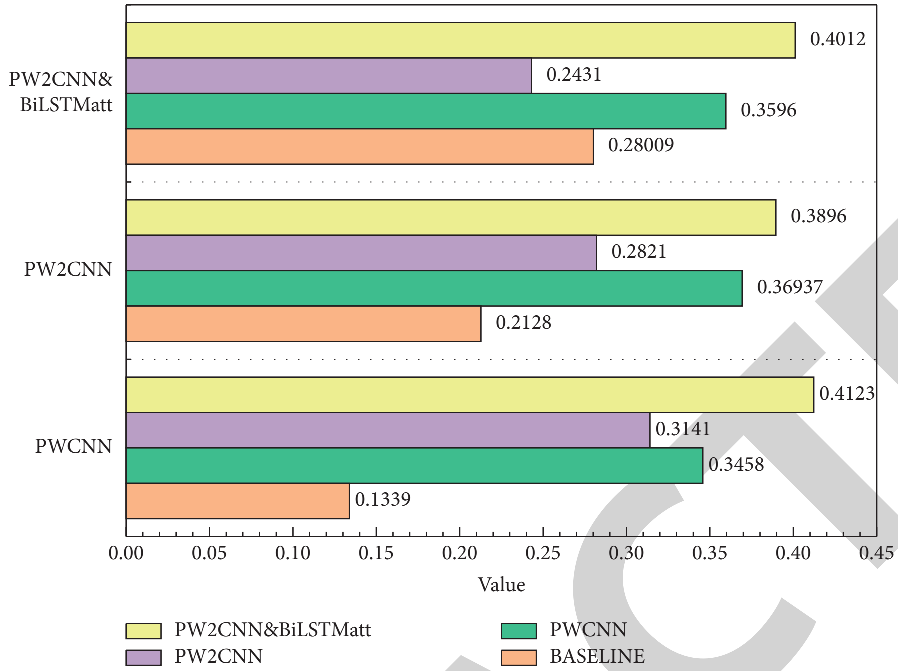

Figure 8: Comparison effect of *t*-test for the four models.

The comparative effect of the *t*-test for the four models is shown in Figure 8.

At a significance level of 0.001, the *p*value of the *t*-test of the PW2CNN&BiLSTMatt model against the BASELINE, PWCNN, and PW2CNN models was less than 0.001. Therefore, there was a difference between the PW2CNN&BiLSTMatt and the BASELINE, PWCNN, and PW2CNN models. Models were all significantly different from each other. In other words, at a significance level of 0.001, the PW2CNN&BiLSTMatt model significantly outperformed the BASELINE, PWCNN, and PW2CNN models. In addition, the *p* value of the *t*-test of the PWCNN model against the BASELINE model (0.003 39) was less than 0.05 at a level of 0.05. Therefore, there was a difference between the PWCNN model and the BASELINE. In other words, the PWCNN is significantly better than the BASELINE model at a significance level of 0.05.

5. Conclusion

A PW2CNN and BiLSTMatt model for sentiment evaluation is proposed for mining and analyzing the sentiment of foreign literary works, using both word vectors and POS feature vectors. These two features are two ways of extracting information from textual data, mainly taking into account the characteristics of words themselves and the characteristics of features containing sentiment information. The role of lexical features in the model is verified by comparing the BASELINE model with the PWCNN model. At the same time, two options are proposed for the fusion of these two features, one is the direct splicing of two vectors and the other is the "fusion" by means of a parallel structural neural network model. A comparison is made between a CNN and a bidirectional long and short-term memory network incorporating an attention mechanism, and experimental results are presented to demonstrate which model is more suitable. The convolutional network is good at capturing local features, while the long and short-term model is more suitable for features containing "temporal" information. Since in the summary of the study of BASELINE and PWCNN models, no comparative experiments with multiple parameters were conducted, but only between models, the study of the effect of multiple parameters on experimental results will be a key direction, in future work.

Data Availability

The dataset can be accessed upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] X. Jin, L. Kumar, Z. Li et al., "A review of data assimilation of remote sensing and crop models," *European Journal of Agronomy*, vol. 92, no. 1, pp. 141–152, 2018.

[2] A. G. Laborte, M. A. Gutierrez, J. G. Balanza et al., "RiceAtlas, a spatial database of global rice calendars and production," *Scientific Data*, vol. 4, no. 1, Article ID 170074, 2017.

[3] J. de Leeuw, A. Vrieling, A. Shee et al., "The potential and uptake of remote sensing in insurance: a review," *Remote Sensing*, vol. 6, no. 11, pp. 10888–10912, 2014.

[4] A. Orynbaikyzy, U. Gessner, and C Conrad, "Crop type classification using a combination of optical and radar remote sensing data: a review," *International Journal of Remote Sensing*, vol. 40, no. 17, pp. 6553–6595, 2019.

[5] Y. Wei, X. Tong, G. Chen, D. Liu, and Z Han, "Remote detection of large-area crop types: the role of plant phenology and topography," *Agriculture*, vol. 9, no. 7, pp. 150–153, 2019.

[6] M. Mosleh, Q. Hassan, and E Chowdhury, “Application of remote sensors in mapping rice area and forecasting its production: a review,” *Sensors*, vol. 15, no. 1, pp. 769–791, 2015.

[7] C. a. Liu, Zx Chen, Y. Shao, Js Chen, T. Hasi, and Hz Pan, “Research advances of SAR remote sensing for agriculture applications: a review,” *Journal of Integrative Agriculture*, vol. 18, no. 3, pp. 506–525, 2019.

[8] L. R. Z., D. Z., Z. H. Mansaray, D. Zhang, Z. Zhou, and J Huang, “Evaluating the potential of temporal Sentinel-1A data for paddy rice discrimination at local scales,” *Remote Sensing Letters*, vol. 8, no. 10, pp. 967–976, 2017.

[9] C. F. Chen, N. T. Son, C. R. Chen, L. Y. Chang, and S. H. Chiang, “Rice crop mapping using sentinel-1a phenological metrics,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 41, no. 6, pp. 863–865, 2016.

[10] D. B. Nguyen and W. Wagner, “European rice cropland mapping with sentinel-1 data: the mediterranean region case study,” *Water*, vol. 9, no. 6, p. 392, 2017.

[11] H. Phan, T. Le Toan, A. Bouvet, L. Nguyen, T. Pham Duy, and M. Zribi, “Mapping of rice varieties and sowing date using X-band SAR data,” *Sensors*, vol. 18, no. 2, pp. 316–320, 2018.

[12] M. Miwa and M. Bansal, “End-to-end Relation Extraction Using LSTMs on Sequences and Tree Structures,” in *Proceedings of the Association for Computational Linguistics: Stroudsburg*, pp. 1105–1116, PA, USA, June2016.

[13] P. N. Dredze, “Named Entity Recognition for Chinese Social media with Jointly Trained Embeddings,” in *Proceedings of the Association for Computational Linguistics: Stroudsburg*, pp. 548–554, PA, USA, July2015.

[14] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, “Event extraction via dynamic multi-pooling CNNs,” *Association for Computational Linguistics: Stroudsburg*, vol. 8, no. 7, pp. 167–176, 2015.

[15] N. Gupta, S. Singh and D. Roth, Entity linking via joint encoding of types, descriptions, and context,” *Association for Computational Linguistics: Stroudsburg*, vol. 12, no. 3, pp. 2681–2690, 2017.

[16] X. Yao and B. Van Durme, “Information extraction over structured data: question answering with freebase,” *Association for Computational Linguistics: Stroudsburg*, vol. 9, no. 6, pp. 956–966, 2014.

[17] J. P. Chiu and E Nichols, “Named entity recognition with bidirectional LSTM-CNNs,” *Transactions of the Association for Computational Linguistics*, vol. 4, no. 8, pp. 357–370, 2016.

[18] P. Cao and Y. Chen, “Adversarial transfer learning for Chinese named entity recognition with self-attention mechanism,” *Association for Computational Linguistics: Stroudsburg*, vol. 9, no. 5, pp. 182–192, 2018.

[19] K. Cho, B. van Merriënboer, and C. Gulcehre, “Learning phrase representations using RNN encoder-decoder for statistical machine translation[J],” *Association for Computational Linguistics: Stroudsburg*, vol. 42, no. 5, pp. 1724–1734, 2014.

[20] H. He and X. Sun, “F-score driven max margin neural network for named entity recognition in Chinese social media,” *Association for Computational Linguistics: Stroudsburg*, vol. 52, no. 2, pp. 713–718, 2017.

[21] N. Peng and M. Dredze, “Improving named entity recognition for Chinese social media with word segmentation representation learning,” *Association for Computational Linguistics: Stroudsburg*, vol. 62, no. 2, pp. 149–155, 2016.

[22] J. Pennington and R. M. C. Socher, “Global vectors for word representation Association for Computational Linguistics Stroudsburg,” *Association for Computational Linguistics: Stroudsburg*, vol. 8, no. 2, pp. 1532–1543, 2014.

[23] V. Mnih and N. Heess, “In Advances in neural information processing systems,” *Red Hook*, vol. 61, no. 3, pp. 2204–2212, 2014.

[24] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *ArXiv*, vol. 8, no. 6, pp. 1409–1416, 2014, https://arxiv.org/abs/1409.0473.

[25] P. S. W. T. Zhou, “J. Attention-based bidirectional long short-term memory networks for relation classiffication. association for Computational Linguistics,” *Stroudsburg*, vol. 6, no. 5, pp. 207–212, 2016.

[26] G. B. Lample and M. Subramanian, “Neural architectures for named entity recognition,” *Human Language Technologies[J]. Association for Computational Linguistics: Stroudsburg*, vol. 8, no. 5, pp. 260–270, 2016.

[27] H. He and X. Sun, “A unifiied model for cross-domain and semi-supervised named entity recognition in Chinese social media[J],” *Proceedings of the Thirty*, vol. 12, no. 2, pp. 3216–3222, 2017.

[28] S. Li, Z. Zhao, and R. Hu, “Analogical reasoning on Chinese morphological and semantic relations,” *Association for Computational Linguistics: Stroudsburg*, vol. 124, no. 3, pp. 138–143, 2018.

[29] A. L.-F. Han, X. Zeng, D. F. Wong, and L. S. Chao, “Chinese named entity recognition with graph-based semi-supervised learning model[J],” *Association for Computational Linguistics*, vol. 6, no. 4, pp. 15–20, 2015.

[30] Y. Zhang and J. Yang, “Chinese NER using lattice LSTM,” vol. 32, no. 8, pp. 84–88, 2018, https://arxiv.org/abs/1805.02023.