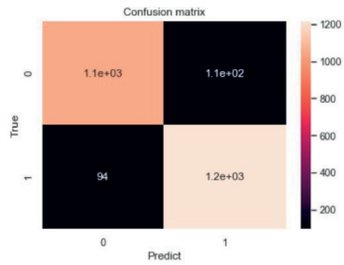
A novel hybrid model by using convolutional neural network and long short-term memory for text sentiment analysis



Un nuevo modelo híbrido de red neuronal convolucional y memoria de largo-corto plazo para el análisis de opinión en texto

Xiaohui Ma

Information Department, Business College of Shanxi University, Suite 24, Unit 2, Building 4, 60 Wangcun South St, Xiaodian District, Taiyuan City, Shanxi Province, China, 030012, belinda.ma1982@gmail.com

DOI: http://dx.doi.org/10.6036/9714 | Recibido: 02/03/2020 • Inicio Evaluación: 03/03/2020 • Aceptado: 13/05/2020

RESUMEN

- El aprendizaje automático se ha aplicado ampliamente en el análisis de las emociones. Sin embargo, el lenguaje natural tiene una relación dependiente de la estructura, confiando en una sola red para la extracción de características, lo que limita la precisión de la clasificación del sentido del texto. Para extraer características semánticas y realizar la clasificación de manera efectiva, se propuso en el estudio la red neural convolucional y el modelo de memoria a largo plazo (CNN-LSTM). Se empleó el método Word2vec para entrenar el vector inicial de la palabra. Se introdujeron la capa convolucional de la CNN y la capa de agrupación máxima para extraer las características locales del texto. Se adoptó el módulo LSTM para capturar las dependencias a largo plazo entre las secuencias de palabras. Las características extraídas con el LSTM y la CNN se fusionaron, y se mejoró la eficiencia utilizando la tecnología de regularización de la deserción. Se realizó un estudio de caso en el conjunto de datos de la revisión de la película para analizar el efecto de la clasificación utilizando los métodos tradicionales y propuestos. Los resultados demuestran que, en comparación con los modelos únicos de CNN y LSTM, el índice de evaluación F1 de este modelo mejora en un 1,93% y un 0,97%, respectivamente. En comparación con el vector de características inicializadas aleatoriamente, la precisión del modelo mejora un 2,74% por la incrustación de palabras. El método propuesto puede recibir la entrada paralela de la información de texto, lo que reduce el tiempo de formación del modelo de red. El mecanismo que combina la CNN y el LSTM compensa las deficiencias de depender únicamente de la incrustación de palabras para la extracción de características, lo que permite al modelo obtener una característica de la opinión y una polaridad de la misma efectivamente identificada sin conocimientos externos como el análisis de sintaxis de dependencia. El método propuesto proporciona una referencia específica para la clasificación de las opiniones en los textos de los comentarios en los medios de comunicación social.
- **Palabras clave:** Sentiment analysis, CNN, LSTM, Feature fon.

ABSTRACT

Machine learning has been widely applied in sentiment analysis. However, natural language has a structure-dependent relationship, relying on a single network for feature extraction, which limits the precision of text sentiment classification. To extract semantic features and perform classification effectively, the convolutional neural network and long short-term memory model (CNN–LSTM) was proposed in the study. The Word2vec method was employed to train the initial word vector. The CNN convolutional layer and the max-pooling layer were introduced to extract the local features of the text. The LSTM module was adopted to capture the long-term dependencies between word sequences. The features extracted with LSTM and CNN were fused, and efficiency was improved using dropout regularization technology. A case study was performed on the movie review dataset to analyze the classification effect utilizing the traditional and proposed methods. Results demonstrate that compared with the single CNN and LSTM models, the evaluation index F1 of this model improves by 1.93% and 0.97%, respectively. Compared with the randomly initialized feature vector, the precision of the model improves 2.74% by the word embedding.The proposed method can receive parallel input of the text information, which reduces the training time of the network model. The mechanism combining CNN and LSTM compensates for the shortcomings of relying only on word embedding for feature extraction, enabling the model to obtain in-depth sentiment feature and effectively-identified sentiment polarity without external knowledge such as dependency syntax analysis. The proposed method provides a specific reference for the sentiment classification of comment texts in social media.

Keywords: Sentiment analysis, CNN, LSTM, Feature fusion.

1. INTRODUCTION

Sentiment Analysis (SA) is an essential task in the field of Natural Language Processing (NLP), which aims to mine the emotions expressed by users. Mining in-depth sentiment information in text and applying deep neural network models to different types of SA tasks have attracted attention from scholars.

The most commonly applied methods to address the SA problem belong either in the rule-based methods or machine learning-based methods. Rule-based methods mainly perform SA on the basis of sentiment dictionaries, feature statistics, and sentiment templates obtained from artificial experiences or expert opinions. Machine learning-based methods manually label the training data set, perform feature extraction, and train a classification model. Finally, unlabeled test samples are input to the classification model for sentiment discrimination. In recent studies, classification models based on machine learning have achieved many breakthrough results in sentence SA tasks

[1-2]. The method based on machine learning has achieved a better classification effect than rule statistics without relying on complex artificial rules and has reduced labor costs. However, such methods rely on external knowledge, such as feature engineering or dependency parsing.

In previous studies, scholars tried to perform SA on product reviews in a single field. The primary method is to train a separate sentiment classification model for each product. In recent years, with the introduction of deep learning, deep neural network models have been widely applied for SA. The mainstream deep neural network models include convolutional neural networks (CNN), recurrent neural networks (RNN), and the long short-term memory (LSTM). CNN extracts the local semantic information of a sentence through a convolution operation, which is the first deep network model employed to perform sentiment classification. Kim et al. [3-4] introduced a novel idea to SA, which involves adopting pre-trained word vector models to vectorize sentence sequences, aiming to extract lexical-to-sentence-level features by CNN's layer. Sequence information is compressed by convolutional and pooling layers and performs sentence-level sentiment classification. However, the problem of losing long sequence semantics remains. The time series models represented by RNN and LSTM have strong sequence modeling capabilities, can alleviate the long-term dependence of sequences, and have achieved significant results in the field of SA [5-9,26-30]. This type of studies mainly introduced LSTM models for sequence modeling, and incorporated Pos information and parameterized it into the network with distributed semantic representations, which improves the classification performance of the algorithm. However, problems arise in effectively identifying critical phrases in sentences, and training time is too long.

Concerning the available measures of SA, existing solutions focus on feature extraction from a single dimension and requiring syntactic dependencies, which affect the performance of the model. CNN can obtain parallel input information to represent the importance of each word in a sentence, and LSTM can capture long sequence semantic information.

On the basis of this analysis, to learn the sentiment polarity of a text, this study proposes a SA model based on multi-feature fusion, aiming at extracting the feature information accurately, and then providing a reference for the related work.

2. STATE OF THE ART

In the existing work, dictionary-based SA methods and machine learning-based supervised learning methods have achieved great success in SA tasks. However, the method based on sentiment dictionary needs numerous rules and has poor generalization ability. The method based on traditional machine learning requires pre-processing and feature engineering for the input text. In recent years, deep learning has made significant breakthroughs in SA. The methods do not require too much pre-processing, which reduces the model's dependence on feature engineering and labor costs. The models can extract emotional features by self-learning. The SA task has a better classification effect than traditional machine learning methods.

In combination with emotional field words and polysemous emotional words, Xu [10] and Xi [11] et al. expanded the emotional dictionary, relying on the emotional dictionary to count the number of positive and negative emotional words. Although scholars have achieved excellent results, the accuracy of SA related to the size of the sentiment dictionary, and which reduced the generalization ability of the model. Bagus setya et al. [12] presented a semantic-based text feature extraction method, which utilizes the Word Sense Disambiguation technology and the algorithm of WordNet similarity to assign the correct sentiment value to each word. The features improved the performance of SA, but the algorithm depended on the domain feature dictionary. In [13], Fang et al. performed SA using support vector machine (SVM), random forest (RF), and Naive Bayes (NB) technologies in the product review data set from Amazon and generated good results. However, the parameters to be estimated increased linearly with the number of documents, which was prone to overfitting. The model developed by Cai et al. [14] adopted the sentiment dictionary to construct structured features of a text and then combined with a hybrid model of SVM and Gradient Boosting Decision Tree to perform text SA, which achieved better classification effect than a single model. However, it employed plenty of artificially structured text features and had a poor real-time performance. LSTM brings the gate mechanism into the traditional RNN, which overcomes the shortcomings of RNN. Many models have been improved on the basic LSTM and GRU models and achieved good classification results. The model developed by Socher et al. [15-16] defined Tree-LSTM to extract semantic features of the text and introduced external memory units in the LSTM to improve the model's ability. However, accuracy improvement was limited due to a mass of external memory matrix parameters. Liang et al. [17] adopted the SA method based on polarity transfer and LSTM neural networks. They applied LSTM to extract the semantic information of the text in the sentiment polarity transfer model, which improved the accuracy of SA. Nevertheless, scholars did not consider the contextual information and local semantic features' effect on SA.

In the study of Liu et al. [18-19], CNN was applied to Weibo comments to obtain the opinion of users and overcome the difficulty of feature extraction. However, it could not consider the relation between sentences, and thus lost the sequent feature of the text. Ruangkanokmas et al. [20] proposed a deep belief network based on feature selection to solve lexical problems. The work processed four different data sets in sentiment classification experiments, comparing feature selection to evaluate the effection and the accuracy of the model, but the training time of the network was too long, making it unsuitable for processing large-scale problems. Yuan et al. [21-22] conducted a comprehensive work on semantic analysis, which extracted features with different sizes of the word embedding matrix by using multi-channel convolution and fused the features as the input of the bidirectional GRU to model the text. The model achieved efficient results with fewer convolutional layers on Stanford Sentiment Tree Bank dataset and IMDB dataset. However, the model was limited to insufficient stability and generalization ability. The work [23] presented by Conneau et al. focused on using deep convolutional neural networks to perform sentence classification, but the convolutional layers required increasing to 29, and the model was complex. Dongmin et al. [24] proposed a convolutional neural network for target-level SA tasks. The algorithm employed distance information between target words and neighboring words to determine the importance of each word to the target word. The approach achieved good performance on a single target dataset but lost location information after the max-pooling layer, which affected the capture of the positional relationship between words. The emotion classification method developed by Yunseok et al. [25] defined two CNNs. The first method extracted the position information of the aspect expression, and the second performed the sentiment classification of the target using the

map. Scholars evaluated the model on the SemEval 2016 Task 5 dataset, showing that the model had a specific practical value. However, the model processed the entire text into a sequence of words without using text hierarchy to set weights, leading to the keywords, and the ordinary words had the same contribution to the final classification.

Existing studies mainly optimized for a single network structure, and most of them focused on local feature information independently. Only a few studies have been done on feature fusion, especially on the correlation of local and long sequence feature fusion. This study proposes the CNN–LSTM method from the perspective of feature fusion, processing local and extended sequences of text information in parallel at the same time, avoiding losing relevant feature information, and guaranteeing the integrity of features at all levels. In the net structure, the study explores the semantic association between texts by using the feature fusion method, which provides a reference for the optimization of text SA.

The remainder of this work is organized as follows. Section 3 introduces the structure, construction, feature fusion, and parameter selection of the model. Section 4 compares the evaluation indexes and analyzes the impact of the initial feature vector on the classification performance. Section 5 summarizes the conclusions.

3. METHODOLOGY

The model proposed in this study took word embeddings as input, inputting vectors to the convolutional layer to extract local features, and to the LSTM model to extract semantic features in long sequences. The extracted features were fused, dropout processing was carried out, and inputted to SoftMax for classification. Fig. 1 shows the architecture of the hybrid CNN–LSTM model. The proposed method improves the effect of text SA by taking advantage of LSTM and CNN.

During the pre-processing stage, sentence segmentation, tagging, stop word removal, and stemming tasks were performed. The model adopted Word2Vec for word embedding, the CNN–LSTM layer for feature extraction and fusion, and took the fused features as the input. The model used the dropout mechanism, thus discarding part of the trained parameters in iteration. As a result, the weight update no longer depended on inherent features, thus preventing overfitting. Finally, the Softmax classifier yielded the results.

3.1 WORD EMBEDDING

Word embedding maps words into a low-dimensional continuous vector. The most typical word embedding model is the Word2Vec proposed by Google's Mikolov. It includes CBOW and Skip-gram models. The CBOW model predicts target words from context words, and the Skip-gram model predicts context words from target words. The common objective of the two models is to train the word vector with the maximum likelihood function.

For the CBOW model, the predicted probability of the t-th word *w(t)* is P_{CBOW}:

$$P_{CBOW} = P(w(t) \mid w(t-k), \cdots w(t-1), w(t+1), \cdots w(t+k)) \quad (1)$$

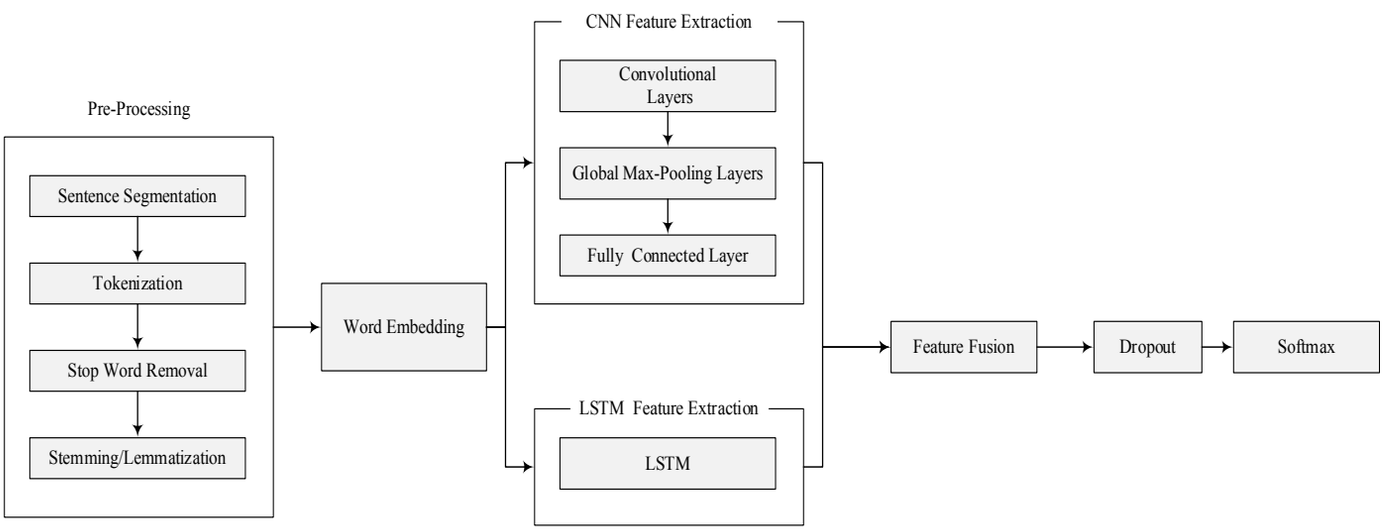


Fig. 1. Model architecture with hybrid CNN–LSTM model

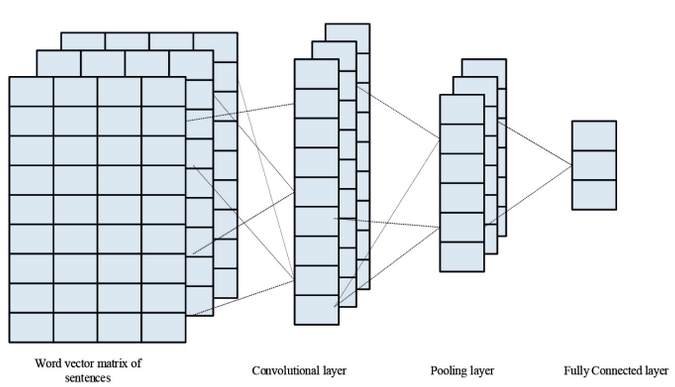


Fig. 2. Architecture of the convolutional neural network

For the Skip-gram model, the predicted probability of the context word of the t-th word *w(t)* is :

$$P_{skip-gram} = P(w(t-k), \cdots w(t-1), w(t+1), \cdots w(t+k)) \mid w(t)) \quad (2)$$

The word vector is updated by calculating the prediction probability and backpropagation. Then the Word2Vec word vector is obtained after a certain number of iterations.

Gensim was employed to train the word vector in the proposed method. Each word was represented as a low-dimensional vector, which can be mapped to a point in space. Words with similar meanings have small distances in space. The word vector sequence

V of the text sequence S is defined as $V = \{v_1, v_2, ..., v_n\}, v_n \in R^m$. Finally, the model stitched V together to obtain the input of the entire model.

3.2 CONVOLUTIONAL NEURAL NETWORK

The convolutional neural network was applied in machine vision in earlier studies, in which in-depth features of images are extracted through convolution operation and the process of rule matching is avoided. In recent years, CNN was commonly adopted in text classification. By mapping text into a matrix composed of word vectors and having a method similar to picture convolution, CNN extracted in-depth features for text analysis. The network structure of CNN can well capture the local features of input data. Fig. 2 shows an architecture of the convolutional neural network.

In natural language processing, a convolution layer utilizes convolution kernels with different window sizes to perform convolution operations on the data. The convolution process involves scanning sample data one by one with the convolution kernel.

The pooling layer processes the results of convolution. Pooling operations include max pooling and average pooling. Max pooling layer selects the maximum value, and the average pooling layer averages the results of convolution. Among them, the max pooling layer can focus on some essential outcomes after convolution while discarding unimportant features, which is similar to feature sampling. The addition of the pooling layer can map the consequence of multiple convolution kernels to the same latitude, reducing the dimension and improving the efficiency of network training.

In NLP tasks, the features of a text are essential information representation. CNN receives the word vector representation through the input layer. The input layer matrix of the network model can be defined as follows:

$$V_s \in R^{n \times d} \tag{3}$$

where d is the word vector dimension, and n is the length of a comment in the sentence.

The convolution layer performs convolution operation on the input matrix with a convolution kernel, in which it extracts local features of the input information, and obtains a convolution feature vector of the input information. Let the length of the convolution kernel be k, and then for the i-th position in the text, the convolution window can be expressed as follows:

$$w_i = \left[v_i, v_{i+1}, ..., v_{i+k-1}\right] \tag{4}$$

When the convolution kernel $w \in R^{k \times d}$ passed through the convolution window, an eigenvalue C_i can be obtained:

$$c_i = f\left(w_i \cdot x + b\right) \tag{5}$$

Through the convolution kernel W, the output C of the convolutional layer can be obtained:

$$c = [c_1, c_2, \cdots c_n] \tag{6}$$

With m convolution kernels, the eigenvector C can be obtained as follows:

$$C = [c^1, c^2, \cdots c^m] \tag{7}$$

To obtain features with semantic information and reduce the parameters of the fully connected layer, the model performs the max pooling operation on the features extracted by the convolutional layer—the maximum value in vector C. Different length information input and different size convolution kernels can extract the output vectors of the same dimension through the pooling layer.

The convolution windows are set to 3, 4, and 5, and the corresponding outputs are C_3, C_4 and C_5, respectively, in the proposed model. Then the model performs the max pooling operation on the feature vectors and obtains a feature vector $p \in R^m$ with a length of m. The output vector of the pooling layer is $[P_3, P_4, P_5]$. The feature vectors stitch to obtain the final text representation vector P, which is transformed into a h_{cnn}.

3.3 LSTM

LSTMs are a unique type of RNN capable of learning long-term dependencies. LSTM is explicitly designed to avoid the long-term dependency problem. LSTM adds gate structures based on RNN, which are the forget gate, input gate, and output gate. Among them, forget gate is selected for long-term memory, input gate controls the input amount of current information, and output gate controls the current output state. In LSTM, where X_t is the original input of the current unit of LSTM, C_t and h_t respectively represent the state and output of the current unit of LSTM, C_{t-1}, and h_{t-1} represent the memory state of the previous unit and the output of a sequence unit on LSTM. Equation 8-10 provides an explanation of LSTM gates and cells. Forget gate in Equation 8, the state value of the current unit in Equation 9, the input gate in Equation 10, and through the forget gate f_t, input gate i_t, and $\widetilde{C}_t$, the model can gain the current unit status value in Equation 11. The output gate and the current unit status value can be employed to calculate output value. Equation 12 shows the output gate formula, and Equation 13 shows the output value.

$$f_t = sigmoid(w_f \cdot [h_{t-1}, x_t] + b_f) \tag{8}$$

$$\widetilde{C}_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \tag{9}$$

$$i_t = sigmoid(w_i \cdot [h_{t-1}, x_t] + b_i) \tag{10}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{11}$$

$$o_t = sigmoid(w_o \cdot [h_{t-1}, x_t] + b_o) \tag{12}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{13}$$

LSTM was employed to process the feature extraction of long-term relationships, and Dropout was employed to avoid overfitting. Then the features were spliced by using the softmax function.

4. RESULT ANALYSIS AND DISCUSSION

4.1 DATA SET AND PARAMETERS

The model was evaluated in Python programming language, and the experiments were implemented in a 64-bit Windows 10 environment (Intel Core i7-7500, GPU GeForce GTX 1050Ti, 8G memory, and Keras).

The study utilized the IMDB movie review dataset [31] to evaluate the proposed hybrid CNN–LSTM model. IMDB dataset having 50K movie reviews for natural language processing or

Parameters	Values
Learning rate	0.001
Dropout	0.5
Embedding dim	100
Optimizer	Adam
Hidden layer nodes	128
Batch size	64
Pooling	Max
Windows size	3,4,5
L2	3

Table I . Model parameters

Text analytics. This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. It provides a set of 25,000 highly polar movie reviews for training and 25,000 for testing. We can get the dataset from http://ai.stanford.edu/~amaas/data/sentiment/ The method contrasted with traditional machine learning and achieved better results in precision, recall, and f1-measure. Table I shows the parameter settings of the hybrid model.

Dropout refers to ignoring units (i.e. neurons) during the training phase of a certain set of neurons which is chosen at random. At each training stage, individual nodes are either dropped out of the net with probability 1-p or kept with probability p, so that a reduced network is left; incoming and outgoing edges to a dropped-out node are also removed.A fully connected layer occupies most of the parameters, and hence, neurons develop co-dependency amongst each other during training which curbs the individual power of each neuron leading to over-fitting of training data.Dropout is an approach to regularization in neural networks which helps to reduce interdependent learning amongst the neurons and to prevent over-fitting. Dropout values will affect the performance of the model. When the value is set as 0.5, the model performance is the best. Figure 3 shows the impact of Dropout on model performance.

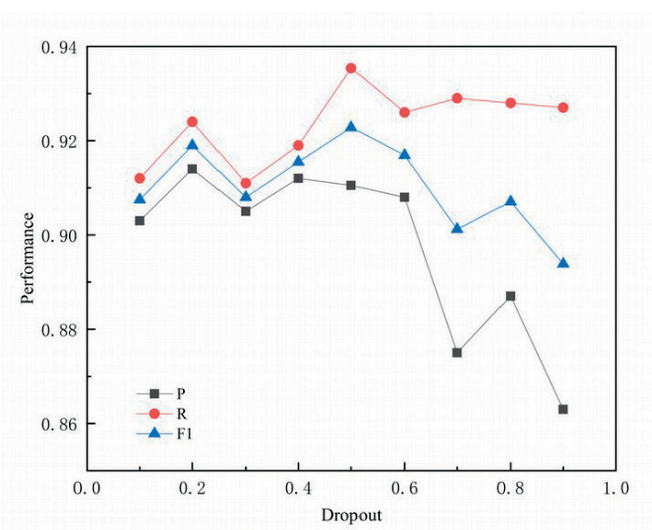


Fig. 3. Impact of dropout value on model performance

4.2 EVALUATION METHODS

The performance of the classification model is evaluated by the following three indices:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (14)$$

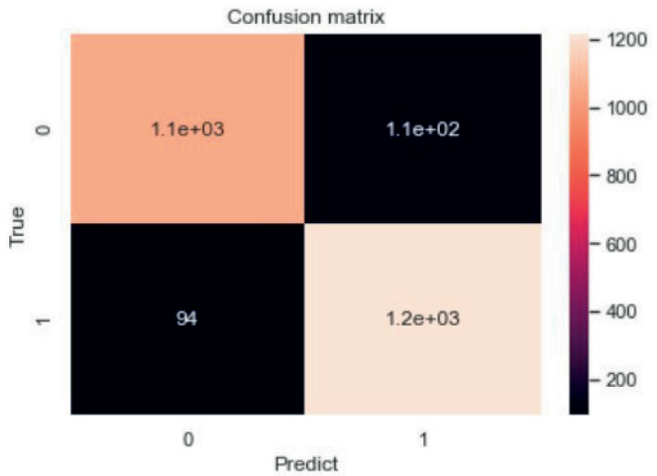


Fig. 4. The confusion matrix of the experiment

$$\text{Recall} = \frac{True\ positive}{True\ positive + False\ Negative} \quad (15)$$

$$\text{F1} - measure = \frac{2 \times (\text{Pr}\, ecision \times \text{Re}\, call)}{\text{Pr}\, ecision + \text{Re}\, call} \quad (16)$$

4.3 RESULTS AND DISCUSSION

The IMDB movie review data is selected as experimental data to evaluate the effectiveness of the text SA method. The performance of the CNN, LSTM, and CNN–LSTM are analyzed in the section.

The study randomly selected 25,000 samples from the IMDB data set and performed a tenfold cross-validation experiment. Figure 4 shows the confusion matrix of the experiment.

Table II shows that the model in the study is better than a single CNN and LSTM SA algorithm model in terms of recall and precision. The evaluation index F1 of the model raised by 1.93% and 0.97% in contrast to the CNN and LSTM, respectively. + indicates that the result of paired t-test is significant at $p < 0.05$.The CNN model only considers the local semantic features, and without the context of the text. On the contrary, the LSTM model only regards the sequence information of the text and without the influence of local semantic features. The model in this study took the advantages of LSTM and CNN models, applying LSTM to extract context features, and CNN to obtain local semantic features, comprehensively considering the influence of local and long-term semantic features on text SA. The proposed model has better performance than the other two algorithms of single-text SA.

Table II also compares the runtime of each iteration of the three methods. The traditional CNN is easy to parallel on the GPU and has the fastest speed. Each unit of the LSTM requires complex calculation operations and the runtime time is longer. The model proposed in this paper uses parallel processing in the feature extraction stage, which improves the classification effect without significantly increasing the training time.

Models	Precision	Recall	F1-measure	Time Cost
CNN	0.9012	0.9056	0.9034	4
LSTM	0.9073	0.9189	0.9131	27
CNN–LSTM	0.9105+	0.9354	0.9228+	29

Table II. Comparison of the proposed CNN–LSTM model with CNN and LSTM on IMDB movie veview dataset

Models	Precision
SVM	0.8402
Bi-directional GRU	0.908
Attention-based LSTM	0.8994
Bag of words	0.8835
C–LSTM	0.9023
CNN–LSTM	0.9105

Table III. Accuracy comparison of the proposed CNN–LSTM model with traditional approaches on the IMDB dataset

The model in the study not only contrasted with the single model but also with other deep learning models. Table III shows the experimental results. In the first model, after the extraction, representation, and normalization of the text, the features are input into the SVM to learn the classification function. It is transformed into a quadratic optimization problem to find the optimal value. The attention-based LSTM model proposed in [32] introduced the attention model to calculate the influence weight of historical nodes on the current node and explored the impact of the context on classification. In [33-34], the bag of words text classification ignored the information of word order, grammar, and each word was independent of each other. The model in [35] first obtained text features by CNN and then inputted features into LSTM for classification. Table III shows that the performance of the proposed model is more effective than the comparison model.

The study randomly selected 5,000 samples from the IMDB data set and performed a tenfold cross-validation comparison experiment to analyze the impact of the initial value on classification performance. Figure 5 shows the comparison results. The model using the Word2vec training feature vector achieves better classification results than the randomly initialized feature vector in all experiments, indicating that using the Word2vev feature vector can effectively improve the accuracy of emotion classification. The analysis results show that unlike the randomly initialized feature vector, using the Word2vec vector can make the network more effective and realize better classification performance. The CNN–LSTM model adopted multi-feature fusion as input not contained not only the information of the feature itself but also the relationship between the features. The interrelationships effectively make up for the inadequacy of the initial value of the feature vector, which reduced the model's dependence on the initial value of the vector.

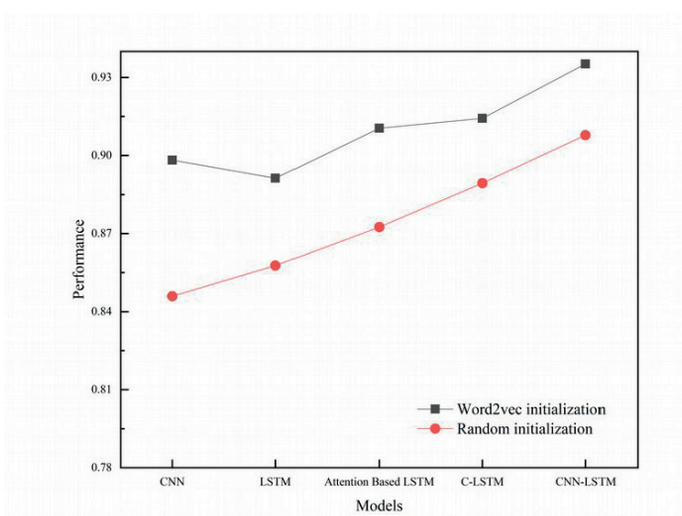


Fig. 5. Comparison results on IMDB

5. CONCLUSION

To avoid losing feature information and improve the performance of SA, a novel method was developed to extract the features of the two models and was combined with the dropout to achieve better classification efficiency. A case study was analyzed over the traditional and proposed methods. The following conclusions could be drawn:

(1) The model depends on the initial value of the feature vector, and the word embedding can effectively increase the classification effect of SA. Results demonstrate that compared with the randomly initialized feature vector, the precision of the model improves 2.74% by the word embedding.

(2) The model can extract more information with the combination of the keywords and long sequence relation features. It can well reveal the significant degree of each word and the relationship between information, which will contribute to the betterment of the proposed hybrid solution.

(3) The introduction of the LSTM module improves the precision of sentiment classification. Results demonstrate that compared with the single CNN and LSTM models, the evaluation index F1 of this model improves by 1.93% and 0.97%, respectively. Integrating syntactic analysis into the current model should be considered in future studies.

Unlike relevant models in the SA field, the classification effect of the model in this study has been significantly improved. However, this model only considered positive and negative binary text SA. Different text language types, aspect-level sentiment classification, and feature fusion strategy will be considered in future work.

REFERENCES

[1] Agarwal B, Mittal N. "Machine Learning Approach for Sentiment Analysis". Socio-Affective Computing 2. Switzerland: Springer International Publishing, 2016. p.21.ISBN:978-3-319-25341-1.DOI: http://dx.doi.org/10.1007/978-3-319-25343-5_3

[2] Tripathy A, Agrawal A, Rath SK. "Classification of sentiment reviews using n-gram machine learning approach". Expert Systems with Applications. March 2016. Vol.57.p.117–126.DOI: http://dx.doi.org/10.1016/j.eswa.2016.03.028

[3] Santos CN, Gatti MA. "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts". in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. August 2014. p.69-78.

[4] Kim Y. "Convolutional neural networks for sentence classification". In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. October 2014. p.1746-1751. DOI: http://dx.doi.org/10.3115/v1/D14-1181

[5] Irsoy O, Cardie C. "Opinion mining with deep recurrent neural networks". In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). October 2014. p.720-728. DOI: http://dx.doi.org/10.3115/v1/D14-1080

[6] Yang DL, Zhu R,Li Y. "Self-Attentive Neural Network for Hashtag Recommendation". Journal of Engineering Science and Technology Review. April 2019.Vol.12-2.p.104-110.DOI: http://dx.doi.org/10.25103/jestr.122.15

[7] Huang M, Xie H, Rao Y, et al. "Sentiment Strength Detection With a Context-dependent Lexicon-based Convolutional Neural Network". Information Sciences. May 2020.Vol.520.p.389-399.DOI:http://dx.doi.org/10.1016/j.ins.2020.02.026

[8] Cui X, Li S, Nan YT. "Improving Affective Computing Based on Semantic Analysis". Journal of Engineering Science and Technology Review. October 2019.Vol.12-5.p.36-42.DOI: http://dx.doi.org/10.25103/jestr.125.05

[9] Yang L, Geng X, Liao H. "A web sentiment analysis method on fuzzy clustering for mobile social media users". Eurasip Journal on Wireless Communications

& Networking. May 2016.Vol.128-2016.p.1-13.DOI: https://doi.org/10.1186/s13638-016-0626-0

[10] Xu G, YU Z, YAO H, et al. "Chinese text sentiment analysis based on extended sentiment dictionary". IEEE Access. April 2019. Vol.7.p.43749-43762.DOI: http://dx.doi.org/10.1109/ACCESS.2019.2907772

[11] Xi YH. "Construction of Domain-specific sentiment Lexicon in Product reviews". Journal of Chinese Information Processing. September 2016. Vol.30-5.p.136-144.

[12] Rintyarna BS,Sarno R,Fatichah C. "Semantic Features for Optimizing Supervised Approach of Sentiment Analysis on Product Reviews".Computers. July 2019.Vol.8-3.p.1-16.DOI: http://dx.doi.org/10.3390/computers8030055

[13] Fang X, Zhan J. "Sentiment analysis using product review data". Journal of Big Data.June 2015.Vol.2.p.1-14.DOI: http://dx.doi.org/10.1186/s40537-015-0015-2

[14] Cai Y,Yang K,Huang D,et al. «A hybrid model for opinion mining based on domain sentiment dictionary». Intonational Journal of Machine Learning and Cybernetics. December 2017.Vol.10-8.p.2131-2142.DOI: http://dx.doi.org/10.1007/s13042-017-0757-6

[15] Socher R, Perelygin A, Wu JY, et al. "Recursive deep models for semantic compositionality over a sentiment treebank". In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. October 2013.p.1631-1642.

[16] Tran K, Bisazza A, Monz C. "Recurrent memory networks for language modeling". In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. June 2016.p.321-331.DOI: http://dx.doi.org/10.18653/v1/N16-1036

[17] Liang J, Chai YM, Yuan HB, et al. "LSTM with recursive networks based on polarity shifting for sentiment analysis". Journal of Chinese Information Processing. May 2015.Vol.29-5.p.152-160

[18] He YX, Sun ST, Nu FF, et al. "Deep learning model enhanced with Emotion Semantics for Microblog sentiment Analysis". Chinese Journal of Computers. April 2017.Vol.40-4.p.773-790.

[19] Liu YM, Chen YD. "Research on Chinese Micro-Blog Sentiment Analysis Based on Deep Learning". In Proceedings of the 2015 8th International Symposium on Computational Intelligence and Design (ISCID). December 2015.p.358-361.DOI: http://dx.doi.org/10.1109/ISCID.2015.217

[20] Ruangkanokmas P, Achalakul T, Akkarajitsakul K. "Deep Belief Networks with Feature Selection for Sentiment Classification". In Proceedings of 2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS). March 2017.p.9-14.DOI: http://dx.doi.org/10.1109/ISMS.2016.9

[21] Yuan HJ, Zhang X, Niu WH, et al. "Sentiment Analysis Based on Multi-channel Convolution and bi-directional GRU with attention mechanism". Journal of Chinese Information Processing. October 2019.Vol.33-10.p.109-117

[22] Li CL, Xu B, Wu GW,et al. "Recursive deep learning for sentiment analysis over social data". In Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT). October 2014.p.180-185.DOI: http://dx.doi.org/10.1109/WI-IAT.2014.96

[23] Conneau A, Schwenk H, Barrault L, et al. «Very deep convolutional networks for text classification». In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics.April 2017.p.1107-1116.DOI: http://dx.doi.org/10.18653/v1/E17-1104

[24] Hyun D, Park C, Yang M, et al. «Target-Aware Convolutional Neural Network for Target-Level Sentiment Analysis». Information Sciences. April 2019. Vol.491.p.166-178.DOI: http://dx.doi.org/10.1016/j.ins.2019.03.076

[25] Noh Y, Park S, Park SB. "Aspect-Based Sentiment Analysis Using Aspect Map". Applied Sciences. August 2019.Vol.9-16.p.3239.DOI: http://dx.doi.org/10.3390/app9163239

[26] Hassan A, Mahmood A. «Deep learning approach for sentiment analysis of short texts». In Proceedings of the 2017 3rd international conference on control, automation and robotics (ICCAR). April 2017 .p.705-710. DOI: http://dx.doi.org/10.1109/ICCAR.2017.7942788

[27] Usama M, Xiao W, Ahmad B, et al. «Deep Learning Based Weighted Feature Fusion Approach for Sentiment Analysis». IEEE Access. September 2019.Vol 7.p.140252-140260. DOI: http://dx.doi.org/10.1109/ACCESS.2019.2940051

[28] Wang X, Jiang W, Luo Z. «Combination of convolutional and recurrent neural network for sentiment analysis of short texts». In Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers. December 2016.p.2428-2437.

[29] Usama M, Ahmad B, Singh AP, et al. «Recurrent Convolutional Attention Neural Model for Sentiment Classification of short text». In Proceedings of 2019 International Conference on Cutting-edge Technologies in Engineering (ICon-CuTE).November 2019.p.40-45. DOI: http://dx.doi.org/10.1109/ICon-CuTE47290.2019.8991445

[30] Liu Y, Ji L, Huang R et al. «An attention-gated convolutional neural network for sentence classification». Intelligent Data Analysis. October 2019. Vol.23.p.1091-1107.DOI: http://dx.doi.org/10.3233/IDA-184311

[31] Maas AL, Daly RE, Pham PT, et al. "Learning Word Vectors for Sentiment Analysis". In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.June 2011. Vol.1.p.142–150.

[32] Zhang C. "Text classification based on Attention-Based LSTM model". Nanjing University.May 2016.p.1-55.

[33] Zhang X, Zhao J, LeCun Y. "Character-level convolutional networks for text classification". In Proceedings of the 28th International Conference neural Information Processing Systems.December 2015.Vol.1.p.649-657

[34] Yang L, Liu JL,Zhang Y."An Intelligent Security Defensive Model of SCADA Based on Multi-Agent in Oil and Gas Fields". International Journal of Pattern Recognition and Artificial Intelligence. January 2020.Vol.34-1.p.1-13.DOI: https://doi.org/10.1142/S021800142059003X

[35] Huang Q,Chen R, Zheng X,et al. "Deep Sentiment Representation Based on CNN and LSTM". In Proceedings of 2017 International Conference on Green Informatics (ICGI). August 2017.p.30-33.DOI: http://dx.doi.org/ 10.1109/ICGI.2017.45

APPRECIATION

This work was supported by the 13th five-year plan of Shanxi Provincial Education Department (Grant No.GH-17097&GH-17100), the soft science research project of Shanxi Province (Grant No.2019041057-1).