Research Article

M-DA: A Multifeature Text Data-Augmentation Model for Improving Accuracy of Chinese Sentiment Analysis

Liya Wang,1 Xinxin Xu,1 Changhui Liu,2 and Zhe Chen2

1College of Artificial Intelligence, Zhejiang Industry and Trade Vocational College, Wenzhou, Zhejiang 325003, China
2College of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430205, China

Correspondence should be addressed to Liya Wang; 1578938844@qq.com

Received 19 January 2022; Revised 16 February 2022; Accepted 9 March 2022; Published 4 April 2022

Academic Editor: Muhammad Usman

Copyright © 2022 Liya Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A neural network based on a word or character embedding is a mainstream model framework in text sentiment analysis and has achieved good results. However, there is a lack of learning about POS-Tagging and Sequence-Tagging. In this research, we propose a multifeature text data-augmentation model (M-DA) with a multiple-input single-output network structure to overcome this problem of Chinese text sentiment analysis. First, this paper sequentially obtains various sequences of Chinese text, including word sequence, pos sequence, char sequence, char_pos sequence, and char_4tag sequence, we use char_pos and the char_4tag to construct a new sequence (4tag_pos) and then use 4tag_pos to mark the characters to obtain the reconstructed characters sequence (char_4tag_pos), so as to achieve the purpose of text enhancement. Then, the Word2Vec method is used to train the initial reconstruction of the character embedding. Finally, the BiLSTM network is used to capture the long-term dependence between the sequences, and the dropout technology and attention are used to improve the accuracy. In the course of the experiment, we also realized that it is better to use the original sequence and the sequence after text enhancement technology as the input of the BiLSTM network. Therefore, our proposed model also discusses the concatenate or dot method to fuse multiple sequences as the final embedding. Multigroup comparison experiments are conducted on the data set, and the results show that the proposed M-DA model is superior to the traditional deep learning technology in terms of accuracy, recall rate, f-measure, and accuracy, and the relative time cost is small.

1. Introduction

Natural Language Processing (NLP) has become an important direction in the field of artificial intelligence, which promotes the continuous development and breakthrough of language intelligence. Text sentiment analysis is the extraction of opinions and tendencies contained in texts with the characteristics of subjective consciousness [1, 2]. It is an important research direction in the field of natural language processing and is widely used in various industries [3]. With the continuous development of the Internet, Weibo, Zhihu, Douban, Tianya Forum, Jingdong, Meituan, Eleme, Taobao, etc., the many APPs and online comment platforms continue to increase. The Internet has deeply affected all aspects of life, and the number of Internet users has reached a high limit. While using the Internet to get information, individuals are also contributing to the creation of that information. Speeches, declarations, and other materials are being shared, and online information is exploding. Through text emotion analysis technology, users’ views or emotional tendencies are mined from a large amount of user data [4]. Consumers can help themselves make purchase decisions according to the emotional tendencies of other users’ comments. Enterprise managers can understand the market demand through the emotional tendency of user comments, to update and improve the products in time. Government personnel can analyze public opinion based on users’ opinions in popular events on social media, to correctly guide the spread of emotions of netizens, effectively control the trend of event development, or provide support for formulating related policies. Therefore, text sentiment analysis has a high
research value. A neural network based on a word or character embedding is a mainstream model framework in text sentiment analysis and has achieved good results. However, there is a lack of knowledge of POS-Tagging, Sequence-Tagging, etc. In this research, we propose a multifeature text data-augmentation model (M-DA) with a multiple-input single-output network structure to overcome the problem of Chinese text sentiment analysis.

The rest of the paper is organized as follows: Section 1 provides the related work, Section 2 provides the text data-augmentation steps and algorithm, Section 3 provides the language preprocessing. The BiLSTM model based on multifeature text data-augmentation is described in Section 4. Experimental analyses and results are explained in Section 5. The conclusion is given in S.

2. Related Work

Text sentiment analysis methods can be divided into two categories, based on sentiment dictionaries and machine learning methods. The method based on the sentiment dictionary is to first use the existing sentiment words in the dictionary to match the words in the sentence and then calculate the sentiment words in the sentence to obtain the comprehensive sentiment tendency of the sentence. Kamps et al. [5] used the WordNet dictionary to conduct word-level sentiment analysis. Hiroshi Kanayama et al. [6] proposed a fully automatic dictionary expansion method for domain-oriented sentiment analysis. However, this method does not consider the connection between words in the text and lacks semantic information, and the classification effect is too dependent on the quality of the dictionary. The successful application of machine learning in text sentiment analysis has effectively promoted the research and development of text sentiment analysis. Traditional machine learning methods require artificially labeled training sets and artificially designed features for emotional feature extraction and then use text classifiers for classification. Commonly used classifiers include naive Bayes, maximum entropy, support vector machines, etc. [7, 8]. Due to the performance of the classifier, the reliance on the number and quality of manually labeled training sets leads to excessive influence of human factors and huge human engineering. Later, methods based on deep learning appeared, and deep learning is an important branch of machine learning.

The recurrent neural network [9] is a popular model, and most studies use it as the method’s basic module. Zhang et al. [10] used distributed word representation technology and RNN network for sentiment classification. Sequential deep learning model LSTM network [11] solves the problem of gradient disappearance or gradient dispersion in the RNN network [12]. BiLSTM [13] is composed of two LSTMs with opposite front and back directions to obtain contextual features. Zhou et al. [14] performed Chinese sentiment analysis by combining Word2Vec and Stacked Bi-LSTM model. In 2017, the Google machine translation team completely abandoned network structures such as RNN and CNN and only used the attention mechanism [15] for machine translation tasks and achieved good results. Luong et al. [16] proposed global and local attention mechanisms, which promoted the application of attention-based models in the field of NLP. Kokkinos et al. [17] proposed self-attention and applied self-attention to sentiment analysis tasks. Part of the research adds the attention mechanism screening feature at the end of the recurrent neural network. Fei et al. [18] combined the Bi-LSTM model with self-attention to form the SA-BiLSTM method. Long et al. [19] studied the sentiment analysis of Chinese text in social media by combining the BiLSTM network with the multi-head attention mechanism (MHAT). It shows that adding attention mechanism screening features can effectively improve the classification effect.

Text representation has been successfully applied to many downstream natural language processing (NLP) tasks as input features, which has a direct impact on the effect of deep learning models [20]. In 2018, Rush et al. [21] first proposed the distributed representation of words used in deep neural language models. Bengio et al. [22] first used neural networks to build language models. Mikolov et al. [23] proposed the Word2Vec [24] technology based on the Log-Bilinear model [25] in 2013, which promoted the rapid development of word vectors. Despite the success and popularity of word embedding, most existing methods use each word as the smallest unit and ignore the morphological information of the word. When optimizing the cost function related to rare words and their context, the rare words cannot be represented well. To solve this problem, recently, Wieting et al. [26] proposed Charagram embedding, in which words or sentences are represented by character n-gram count vectors. This is an easy way to learn character-based combination models to embed text sequences. Sun et al. [27] proposed two new models to build better word representations by modeling external context and internal morphemes in a joint prediction method, called BEING and SEING. These two models can also be extended to learn phrase representations based on distributed morphology theory. Rei et al. [28] proposed a new architecture for combining alternative word representations. In the framework of sequence labeling, both character-level and word-level embedding are used. Rezaeinia et al. [29] showed that the improved word vector (IWW) is very effective for sentiment analysis. Rahimi et al. [30] proposed two new unsupervised models that integrate word polarity information and word cooccurrence into more tailored sentiment analysis features. Word polarity and cooccurrence are clustered together in the form of tension and tension factorization to generate word embeddings. Unlike other alphabet writing systems, the target of Chinese is three different levels of granularity of radicals, characters, and words. Yu et al. [31] proposed a method of joint embedding of Chinese words and their characters and radical subcharacter components and quantitatively evaluated the quality evaluation and word analogy tasks of word embedding learned by the model on word similarity. Peng et al. [32] inspired by aggressive hierarchical embedding [33] designed two fusion mechanisms to merge three granularities and achieved good results on Chinese
sentiment analysis tasks. In addition, Wu et al. [34] proposed using dictionary embedding and polarity reversal for sentiment analysis.

A Chinese character can be a word or part of a multi-syllable word. However, in the above method, the vector calculation that is used to represent the document does not consider the position or part of speech of the words that are composed. So in this work, we propose a multifature text data-augmentation model (M-DA). First, this paper sequentially obtains various sequences of Chinese text, including word sequence (word), part-of-speech sequence (pos), character sequence (char), word part-of-speech sequence (char_pos), and character position sequence (char_4tag), and then uses character position. The part-of-speech sequence constructs a new sequence (4tag_pos) and then uses 4tag_pos to mark the characters to obtain the reconstructed characters sequence (char_4tag_pos), to achieve the purpose of text enhancement. For example, after “十分” is divided into characters, “＋” will be marked as the first character E_m of the quantifier, and “十分” will be marked as the last character E_m of the quantifier. Then, the Word2Vec method is used to train the initial reconstruction of the word embedding. Finally, the BiLSTM network is used to capture the long-term dependence between the sequences, and the dropout technology and attention mechanism are used to improve the accuracy. In the course of the experiment, we also realized that it is better to use the original sequence and the sequence after text enhancement technology as the input of the BiLSTM network. Therefore, our proposed model also discusses the concatenate or dot method to fuse multiple sequences as the final embedding, to get our final multifeature text data-augmentation model (M-DA).

3. Text Data-Augmentation

Text Data-Augmentation is also called data-augmentation, which means that the value of limited data is equivalent to more data without a substantial increase in data. Data enhancement technology is already a standard configuration in the image field, and data enhancement is achieved through techniques such as image flipping, rotation, mirroring, and Gaussian white noise. In the field of NLP, Wei et al. [35] introduced NLP data enhancement technology, proposed an EDA model, and showed that the data-augmentation can prevent overfitting and improve the generalization ability of the model.

This paper considers that the knowledge of text POS-Tagging and Sequence-Tagging is usually very useful, so this paper proposes to use the position and part-of-speech sequence of the characters in the text to reconstruct the original character sequence, to achieve the purpose of data enhancement. Before obtaining the position and part-of-speech sequence (4tag_pos) of a word, it is necessary to obtain the word sequence (word), part-of-speech sequence (pos), character sequence (char), word part-of-speech sequence (char_pos), and character position sequence (char_4tag). Finally, use 4tag_pos to reconstruct char as a new text sequence (char_4tag_pos).

Example: “It’s really good. Breakfast makes me very satisfied.” The text data-augmentation process is shown in Table 1.

The process of text data-augmentation is shown in Table 1. First, the posseg method in jieba is used to obtain POS at the same time as word segmentation. Then the words are divided into characters, and in the same way, the POS is iterated one by one to obtain the POS of the character (char_pos). Regarding text Sequence-Tagging, 2tag over-simplifies the model and cannot obtain sufficient information; 6tag makes the model complicated. It is good if the training sample is large enough, but the training set is too small to obtain accurate information through 6tag [36]. Therefore, this paper uses 4tag for marking. ‘B’ represents the beginning position of the word, ‘M’ represents the middle position of the word, ‘E’ represents the end position of the word, and ‘S’ represents a single character. By iterating the words, mark the characters one by one with the mark 4tag to generate a new text sequence (char_4tag). Then, the char_pos and char_4tag are merged to obtain the position and POS information corresponding to the character (4tag_pos). Finally, we use 4tag_pos to mark the characters to get the reconstructed text (Char_4tag_pos). The position-POS vector corresponding to the character will always follow the character, which can strengthen the semantic logic of the model. The enhanced dependencies in the text are shown in Figure 1. The quantifier “＋十分” (very) can highlight the main morphological features of things. The “B” marked character “＋” is the beginning of the word, and the “十分” marked character B belongs to the part of the word and the end position.

4. Language Preprocessing

Google’s open-source tool Word2Vec converts text strings into numeric vectors, calculates the distance between words, and groups similar words according to their meanings. The CBOW model uses surrounding words to predict the center word, to use the prediction result of the center word, and to use the Gradient Descent method to continuously adjust the vector of the surrounding words. After the training is completed, each word will be used as the central word, and the word vectors of the surrounding words are adjusted so that the word vectors of all words in the entire text are obtained.

This paper uses the CBOW model based on Negative Sampling to construct the above seven sequence vectors. Take, for example, the vectorization of reconstructed text. Given the context of the reconstructed character w, w needs to be predicted. Therefore, for a given Context(w), the reconstructed character w is a positive sample, and other reconstructed characters are negative samples. Assuming that a negative sample subset NEG(w) ≠ 0 about Context(w) has been selected, and ∀w ∈ NEG(w) defines L∞(w) ∈ {1, 0} to represent the label of the word w. The label of positive samples is 1, and the label of negative samples is 0.

For a given positive sample (Context(w), w), the optimization goal of this model is to maximize the posterior probability of a given text.
TABLE 1: The process of text data-augmentation.

| Word  | POS | Char | char_pos | char_4tag | 4tag_pos | Char_4tag_pos |
|-------|-----|------|----------|-----------|----------|----------------|
| 非常  | d   | 真   | d        | B         | B_d      | 真B_d         |
| 真是  | d   | 是   | d        | E         | E_d      | 是E_d         |
| 好    | a   | 好   | a        | S         | S_d      | 好S_d         |
| 真是  | d   | 是   | d        | S         | S_d      | 真S_d         |
| 好    | x   | 好   | x        | S         | S_x      | 好S_x         |
| 真是  | d   | 是   | d        | S         | S_d      | 真S_d         |
| 好    | x   | 好   | x        | S         | S_x      | 好S_x         |

5. The BiLSTM Model Based on Multifeature Text Data-Augmentation

BiLSTM network is one of the commonly used neural networks in text tasks. This paper combines the proposed multifeature text data-augmentation model (M-DA) with the BiLSTM network to complete Chinese sentiment analysis tasks. The M-DA-BiLSTM model is a multiple-input single-output network structure. The schematic diagram of the model is shown in Figure 2.

For a given text sentence, the reconstructed text method to obtain three types of text, such as word, char, Char_4tag_pos, and input the three types of text into the Word2Vec model for training and obtain the word vector $w$, $w \in R^{vocab}$, and the character vector $c_i, c_i \in R^{vocab}$, and reconstruct the character vector $c_i, c_i \in R^{vocab}$, where $n, n', n''$ is the vocabulary size (dict_len), $k, k', k''$ is the vector dimension (vec_dim), and the obtained dictionary index is used to represent the sentence $S$. There are 3 ways of representation:

$$S_1 = \{...a_i,a_j,...\},$$
$$S_2 = \{...b_i,b_j,...\},$$
$$S_3 = \{...c_i,c_j,...\}.$$  

5.2. Embedding Layer. In this paper, we use the pretrained Word2Vec model for word embedding. The preprocessed data set provides a unique and meaningful sequence of words; each word has a unique ID. The embedding layer uses Word2Vec pretraining weights to initialize embedding weights and introduces external semantic information, which is often very helpful to the model. The input of an embedding layer should be a series of integer sequences, and all integers in the sequence will be replaced by corresponding columns in the corresponding word vector matrix, which is its word vector. For example, the word vector represents the sentence $S$ as shown in Figure 3.

After passing through the dropout layer, change the mode of unified learning of weights and unified update of parameters in the previous network. In each training iteration, some parameters in the network are learned.

5.3. Concatenate Layer. Connect the layers of the input list. It takes a list of tensors as input, all shapes are the same except for the connected axis, and it returns a tensor concatenated as all inputs. The sentence $S$ output at time $t$ is shown in Figure 4.

5.4. BiLSTM Layer. It is a combination of forwarding LSTM and backward LSTM. The memory unit in LSTM is controlled by the forget gate $f_t$, the memory gate $i_t$, the temporary memory state $c_t$, the current memory state $c_t$, and the output gate $o_t$, and they are calculated by the hidden state $h_{t-1}$ at the last moment and the current input $x_t$. See equations (5)–(7). In the calculation, the first hidden state needs to be used, but it does not exist and is generally set to a 0 vector in reality.
Figure 2: The M-DA-BiLSTM model.

Figure 3: The word vector represents the sentence S.

Figure 4: Sentence S output at time t.
\[
\begin{bmatrix}
W_f \\
W_i \\
W_c \\
W_o \\
U_f \\
U_i \\
U_c \\
U_o \\
b_f \\
b_i \\
b_c \\
b_o
\end{bmatrix}
\begin{bmatrix}
f_t \\
i_t \\
\bar{c}_t \\
o_t
\end{bmatrix}
= \begin{bmatrix}
\text{logistic} \\
\text{logistic} \\
\tanh \\
\text{logistic}
\end{bmatrix}
\begin{bmatrix}
W_f \\
W_i \\
W_c \\
W_o \\
U_f \\
U_i \\
U_c \\
U_o \\
b_f \\
b_i \\
b_c \\
b_o
\end{bmatrix}
\begin{bmatrix}
s_t \\
h_{t-1}
\end{bmatrix}
+ \begin{bmatrix}
b_f \\
b_i \\
b_c \\
b_o
\end{bmatrix}.
\]
(5)

\[
C_t = i_t \cdot \bar{C}_t + f_t \cdot C_{t-1},
\]
(6)

\[
h_t = o_t \cdot \tanh(C_t).
\]
(7)

\(W, U\) is the weight matrix, \(b\) is the bias vector, and \(\tanh, \text{logistic}\) are the activation functions.

At time \(t\), the hidden state output by the forward LSTM is \(h_t\), and the hidden state output by the backward LSTM is \(\tilde{h}_t\); then the hidden state \(h_t\) output by the BiLSTM is merged by the output of the forward and backward LSTM, as shown in

\[
h_t = \overline{h_t} \oplus \tilde{h}_t.
\]
(8)

5.5. Attention. This paper uses attention to express the correlation between the words in the text sentence and the output result. First, generate the target attention weight \(v_t\), then use the softmax function for probabilism of the attention weight to generate a probability vector \(p_t\). Finally, the generated attention weight is assigned to the corresponding hidden layer state \(h_t\). Different weights are applied to the state at each moment to ensure that the problem of information redundancy is solved while retaining valid information.

\[
v_t = \tanh(W \cdot h_t + b),
\]

\[
p_t = \frac{\exp(v^\top \cdot v_t)}{\sum \exp(v^\top \cdot v_t)},
\]
(9)

\[a_t = \sum p_t h_t.
\]

5.6. Dense Layer. The parameter is 1, and the activation function is the sigmoid function. \(h_w(x_t)\) is the probability that the predicted input \(x_t\) belongs to category 1, as shown in

\[
h_w(x_t) = p(y_t = 1 | x_t, w) = \text{sigmoid}(w^\top x_t)
\]
(10)

In the above equations, \(x_t\) is the sample feature vector, \(y_t \in [0, 1]\) is the true label of the sample, and \(w\) is a trainable parameter.

The model uses a logarithmic loss function to update the parameter weight matrix \(w\) to complete the experience minimization, as shown in

\[
L(w) = \frac{1}{N} \sum_{i=1}^{N} (y_i \log h_w(x_i) + (1 - y_i) \log(1 - h_w(x_i))).
\]
(11)

The summary function is used to print the M-DA-BiLSTM model, as shown in Figure 5.

6. Experiments

The experimental data is a commonly used and public data set in this field, [37], the data set is Chinese shopping review text with binary sentiment tags, and the tag \([0,1]\), where negative emotions are 0 and positive emotions are 1. Data evaluation objects include multiple types, including hotels, milk, books, and mobile phones. The division of the experimental data set is shown in Table 2.

In the experiment section, we will discuss experimental settings, evaluation indicators, comparison methods, and result analysis in detail.

6.1. Experimental Settings. The experimental environment configuration data are shown in Table 3.

Parameter settings will directly affect the classification effect of subsequent models. The specific parameter settings are shown in Table 4.

6.2. Evaluation Indicators. The four model evaluation indicators of accuracy, precision, recall, and F1 are commonly used standards for NLP model evaluation. Accuracy is an evaluation of the correct classification ability of the model. The higher the accuracy, the better the classification ability of the model. Precision is accuracy, and recall is an evaluation of recall. F1 is a weighted average of precision and recall and is a comprehensive evaluation index. The larger the value, the better the model, as shown in equations (12)-(15).

\[
\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}
\]
(12)

\[
\text{Precision} = \frac{TP}{TP + FP}
\]
(13)

\[
\text{Recall} = \frac{TP}{TP + FN}
\]
(14)

\[
F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]
(15)

In the above formula, its specific meaning is shown in Table 5. TP represents positive emotions labeled by the model, which are positive emotions. FP is the positive emotion labeled by the model, which is negative. TN is a negative emotion labeled by the model, which is a negative emotion. FN represents the negative emotion labeled by the model, which is positive.
6.3. Comparison Methods. The experiment uses the deep learning mainstream network BiLSTM and the classical language pretraining technology Word2Vec as the baseline model and sets the following series of comparative experiments.

(1) Word-BiLSTM [14]: The mainstream method BiLSTM network is used for sentiment analysis, and the model uses Word2Vec technology to train word vectors.

(2) Word-BiLSTM-attention [15, 18]: Compared with the model (1), the attention mechanism is conducive to the selection of important features.

(3) Char-BiLSTM-attention [26, 28]: For characters, the language pretraining models are used to convert characters into vector representations, which is also a popular method.

(4) Char_4tag_pos-BiLSTM-attention: The reconstructed text (Char_4tag_pos) proposed in this paper was used as input.

(5) (Char_4tag_pos: Word)-BiLSTM-attention: It is a network structure with multiple inputs and a single output. The input is the Char_4tag_pos vector and...
the \textit{word} vector, which are merged using the concatenate method and then input into the BiLSTM network.

(6) (Char\_4tag\_pos Dot Word)-BiLSTM-attention: Compared with model (5), the results of Char\_4tag\_pos vector and word vector using the Dot method are input into the BiLSTM network.

(7) (Char: 4tag\_pos: Word)-BiLSTM-attention: It is a network structure with three inputs and a single output. The inputs are Char, 4tag\_pos, and word vector, and then the concatenate method is used to merge the three, so that the overall model can obtain more text representation information.

(8) (Char\_4tag\_pos: Word: Char)-BiLSTM-attention: Compared with model (7), the inputs are the Char\_4tag\_pos, Char, and word vectors, and then the concatenate method is used to merge the three. Observe that the Char\_4tag\_pos vector is better than the 4tag\_pos vector. This model is also the final model (M-DA-BiLSTM) proposed in this paper.

In this work, we propose a BiLSTM model (M-DA-BiLSTM) enhanced based on multifeature text data. In addition, the model uses dropout technology and attention mechanism to improve accuracy. The model uses text 4tag sequence and pos sequence to reconstruct char to generate a new sequence (Char\_4tag\_pos) to achieve the purpose of text data enhancement for char. Here, we need to consider the following three situations.

The first is whether the text data enhancement method proposed in this paper has a positive effect when applied to the experimental data set. So we set up experimental groups 1, 2, 3, and 4, send the Char\_4tag\_pos vector directly to the network, and compare it with the mainstream model that uses word vector or char vector as input.

The second is whether the effect of using the enhanced text data sequence and the original sequence as input together is better than a single input. And what is the difference between different fusion methods? So we set up experimental groups 5 and 6, using concatenate or dot method to fuse Char\_4tag\_pos vector and word vector as input.

Third, what impact will more inputs and more features have on the model? And how do different subelements affect the model? So we set up experimental groups 7 and 8, using concatenate to fuse 4tag\_pos, word and char and compare and fuse Char\_4tag\_pos, word, and char.

6.4. Result Analysis. On the test set, the four model evaluation indicators of accuracy, Precision, recall, and F-1 are used to evaluate the model after the training. The results are shown in Table 6, unit (%).

In Table 6, The Word-BiLSTM model is the mainstream model of sentiment analysis [14]. On its basis, The Word-BiLSTM-attention model uses the attention mechanism proposed in [15] to optimize the model. The Chinese have character-level granularity, which is different from English. The Char-BiLSTM-attention model uses the character-level vector proposed in [26] as the input of the neural network BiLSTM. By comparing with 3 groups of popular models, the Char\_4tag\_pos-BiLSTM-attention model has scores of 91.13% and 91.07% on the two comprehensive indicators accuracy and F1, which is better than the control group. The experiment verifies the feasibility and effectiveness of using the reconstructed word vector as the input of the BiLSTM network.

The models in Table 7 are all network structures with multiple inputs and a single output so that the model can obtain more textual information. The first two groups of models are the dual-input structure of Char\_4tag\_pos and word vector. They use two different methods to merge the two types of vectors. By comparing these two different merging methods, we can know that the concatenate method is 91.53% better than the dot method on the accuracy, and the dot method is 91.43% better than the concatenate method on F1. Therefore, Figure 6 shows the time cost of all comparison models. The abscissa (x-axis) is the epochs of iterations when training the model, and the ordinate (y-axis) is the time of each iteration, in seconds (s). It can be seen from Figure 6 that model 1 without attention has the smallest time cost, followed by single-input models 4, 2, and 3, then dual-input models 5 and 6, three-input models 8, 7. The dual-input model based on the dot method has the greatest time cost. Comprehensive analysis shows that the scores of models 5 and 6 are the same inaccuracy. However, the time cost of the dot method is about three times that of concatenate method, so concatenate method is more practical than the dot method.

Compared with model 4, models 5 and 6 are better than the single-input structure of reconstructed word vector inaccuracy, which shows that the dual input of Char\_4tag\_pos and word vector can further optimize the model. Based on this, Char\_4tag\_pos is composed of the basic elements char and 4tag\_pos. So, we discussed the three inputs of char, 4tag\_pos, and word vector and used the merge method concatenate, which is less expensive in time.

By comparing model 7 with models 5 and 6, we can know that such a three-input model 7 achieves scores of 92.35% and 92.25% on the two comprehensive indicators accuracy and F1, and the results are better. The average time is only increased by about 20 s/epoch compared to the dual-input model 5, and it is much shorter than that of model 6.

Model 8 is the final model proposed in this paper. It is a three-input single-output network structure. The model scores 92.35\%, 93.87\%, and 92.25\% on accuracy, precision, and F-1. The training model takes an average of 102 s/epoch. Compared with model 7, the reconstructed character vector is replaced with the 4tag\_pos vector as an input item, and the concatenate merging method can be used to further optimize the model under the premise that the time cost is as small as possible.

Figure 7 shows the distribution of the accuracy rate (val_acc) and the change of the loss rate (val_loss) of the 8 groups of comparative models on the validation set. Figure 7(a) uses a box plot to show the distribution of val_acc values during the entire model training process. The abscissa is the comparison model number, which has the
same meaning as the label in Figure 6, and the ordinate is the val_acc value. The biggest advantage is that it is not affected by outliers and can describe the discrete distribution of data in a relatively stable way. From Figure 7(a), model 4 is compared with the first three groups of models. In the single-input model, the reconstructed word vector contains the highest value as the input. Comparing models 5 and 6 with the first four groups, the median of the dual-input model is higher, and model 5 contains the highest value. Model 8 contains the highest value of val_acc and the highest median value, with a concentrated value distribution.

Figure 7(b) uses a line graph to show the change of val_loss during the training model. The abscissa is the number of iterations when the model is trained, and the ordinate is the loss rate predicted by the model on the validation set after each round of training. For the model loss rate, the smaller the better. From the visual point of view in the figure, model 1 has the highest loss rate, model 6 has the largest fluctuation, and the remaining 6 groups are more concentrated. Model 8 achieves the minimum value in the fourth iteration, and the curve is smooth. Based on the analysis in Figures 7(a) and 7(b), we can know that model 8 has the characteristics of fast convergence, high accuracy, and stability.

Figure 8 counts the prediction results of 8 groups of models on 2221 test samples from six indicators such as TP, FP, TN, FN, Right, and Wrong. The output result is the probability that the predicted sample is 1. For the
Figure 7: The change of Val_acc and Val_loss.

Figure 8: Continued.
convenience of statistics, the output result greater than 0.5 is classified as 1, and the rest are 0. In Figure 8(a), TP represents the correct number of positive samples predicted by the model, and TN represents the correct number of negative samples predicted by the model. The higher the value, the better. In Figure 8(b), FP represents the number of negative sample errors predicted by the model, and FN represents the number of positive sample errors predicted by the model. The lower the value, the better. In Figures 8(c) and 8(d), Right is the number of samples that the model predicts correctly, and Wrong is the number of samples that the model predicts incorrectly. That is, Right = TP + TN, Wrong = FP + FN.

It can be seen from Figures 8(a) and 8(b) that, in the positive sample, model 3 performs better; in the negative sample, model 8 performs better. From the comprehensive indicators Right and Wrong in Figure 8(c) and 8(d), it can be seen that model 8 performs better on the entire sample. Model 8 is also the final model (M-DA-BiLSTM) proposed in this paper.

In summary, for the data set used in this paper, we verify the proposed M-DA-BiLSTM model. Here, we considered the following three situations:

In the first, we set up experimental groups 1, 2, 3, and 4, send the Char_4tag_pos vector directly to the network, and compare it with the mainstream model that uses word vector or char vector as input. It can be seen that the Char_4tag_pos vector is more appropriate on the experimental data in this paper, which indicates that the text data enhancement method proposed in this paper has a positive effect when applied to the experimental data set.

In the second, based on the results of experimental groups 1, 2, 3, and 4, we set up experimental groups 5 and 6, using concatenate or dot to fuse the Char_4tag_pos vector and the word vector as input. It can be seen that the effect of using the text data-enhanced sequence and the original sequence as the input is better than a single input, but the accuracy of the two fusion methods is equivalent, so we have analyzed the time cost, and the results show that the time cost of the concatenate method is small.

In the third, based on the results of experimental groups 5 and 6, we set up experimental groups 7 and 8, using concatenate to fuse 4tag_pos, word, and char, and compare and fuse Char_4tag_pos, word, and char. It can be seen that three-input is better than two-input, and Char_4tag_pos is more suitable as a subelement. However, the time cost of three inputs is nearly twice that of dual inputs. Therefore, training models with more inputs and more features will require higher and higher experimental hardware and software configuration.

7. Conclusion

In this work, we proposed a multifeature text data-augmentation model (M-DA). First, this work sequentially gathers several Chinese text sequences, including word, pos, char, char pos, and char 4tag (which we employ), and char pos produces a new text sequence (4tag pos). To achieve the goal of text enhancement, utilize 4tag pos to mark the characters and get the reconstructed characters sequence (char 4tag pos). Then, the Word2Vec method is used to train the initial reconstruction of the characters embedding. Finally, the BiLSTM network is used to capture the long-term dependence between the sequences, and the dropout technology and attention are used to improve the accuracy. In the course of the experiment, we realized that it is better to use the original sequence and the sequence after text enhancement technology as the input of the BiLSTM network. Therefore, the proposed model compares the experimental results of the concatenate or dot method and chooses to use concatenate. The method fuses multiple sequences as the final embedding, thereby further
improving the accuracy of text classification. This paper focuses on binary polarity detection in sentence-level sentiment analysis. In the future, we recommend investigating the effectiveness of our proposed M-DA-BiLSTM for other sentiment analysis tasks (such as aspect level or multiple sentiment analysis).

Data Availability

The data used to support the findings of this study are included in the article.

Conflicts of Interest

The authors declare that they have no competing interests.

References

[1] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment classification using machine learning techniques,” in Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, pp. 79–86, Stroudsburg, PA, USA, July 2002.

[2] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” Foundations and Trends in Information Retrieval, vol. 2, no. 1-2, pp. 1–135, 2008.

[3] B. Liu, Sentiment Analysis and Subjectivity. Handbook of Natural Language Processing. N. Indurkhya and F. J. Damerau, Eds., Taylor and Francis Group, Boca, FL, USA, Second edition, 2010.

[4] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp. 168–177, Seattle WA USA, August 2004.

[5] J. Kamps and M. Marx, “Words with attitude,” in Proceedings of the First International Conference on Global WorldNet, pp. 332–341, CIIIL, Mysore India, January 2002.

[6] H. Kanayama and T. Nasukawa, “Fully automatic lexicon expansion for Domain Oriented sentiment analysis,” in Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 355–363, Association for Computational Linguistics, Stroudsburg, PA, USA, July 2006.

[7] B. Pang, L. Lee, and A Sentimental Education, “Sentiment analysis using subjectivity summarization based on minimum cuts,” in Proceedings of the Forty Second Annual Meeting on Association for Computational Linguistics, pp. 271–278, Association for Computational Linguistics, Stroudsburg, PA, USA, July 2004.

[8] S. Liu, F. Li, F. Li, X. Cheng, and H. Shen, “Adaptive co-training SVM for sentiment classification on tweets,” in Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management, pp. 2079–2088, ACM, San Francisco CA USA, October 2013.

[9] A. Graves, A. R. Mohamed, and G. Hinton, “Speech Recognition with Deep Recurrent Neural networks,” in Proceedings of the Acoustics, Speech, and Signal Processing, 1988. ICASSP-88. 1988 International Conference on, vol. 38, New York, NY, USA, April 1988.

[10] Y. Zhang, Y. Jiang, and Y. Tong, “Study of sentiment classification for Chinese microblog based on recurrent neural network,” Chinese Journal of Electronics, vol. 25, no. 4, pp. 601–607, 2016.

[11] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] K. Cho, B. V. Merrienboer, C. Gulcehre et al., “Learning phrase representations using RNN Encoder Decoder for statistical machine translation,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734, October 2014.

[13] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997.

[14] J. Zhou, Y. Lu, H. N. Dai, H. Wang, and H. Xiao, “Sentiment Analysis of Chinese Microblog Based on Stacked Bidirectional LSTM,” IEEE Access, vol. 7, p. 1, 2019.

[15] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention Is All You Need,” 2017, https://arxiv.org/abs/1706.03762.

[16] M. T. Luong, H. Pham, and C. D. Manning, “Effective Approaches to Attention-Based Neural Machine Translation,” in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421, September 2015.

[17] F. Kokkinos and A Potamianos, “Structural attention neural networks for improved sentiment analysis,” in Proceedings of the Fifteenth Conference of the European Chapter of the Association for Computational Linguistics, pp. 586–591, EACL, Valencia, Spain, January 2017.

[18] R. Fei, Y. Zhu, and Q. Yao, “A deep learning method based self-attention and Bi-directional LSTM in emotion classification,” Journal of Internet Technology, vol. 21, no. 5, pp. 1447–1461, 2020.

[19] F. Long, K. Zhou, and W. Ou, “Sentiment analysis of text based on bidirectional LSTM with multi-head attention,” IEEE Access, vol. 7, no. 99, p. 1, 2019.

[20] Q. Liu, M. J. Kusner, and P. Blunsom, “A Survey on Contextual Embeddings,” 2020, https://arxiv.org/abs/2003.07278.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” Cognitive Modeling, vol. 10, no. 5, pp. 533–536, 1986.

[22] Y Bengio, H Schwenk, and J. S Senecal, “A neural probabilistic language model,” Journal of Machine Learning Research, vol. 3, no. 6, pp. 1137–1155, 2003.

[23] T. Mikolov, K. Chen, G. Corrado, and H. Veisi, “Efficient Estimation of Word Representations in Vector Space,” 2013, https://arxiv.org/abs/1301.3781.

[24] T. Hayashi and H. Fujita, “Word embeddings-based sentence-level sentiment analysis considering word importance,” Acta Polytechnica Hungarica, vol. 16, no. 7, pp. 7–24, 2019.

[25] A. Mnih and G. E. Hinton, “A scalable hierarchical distributed language model,” in Proceedings of the Twenty First International Conference on Neural Information Processing Systems, pp. 1081–1088, NIPS, Vancouver British Columbia, Canada, December 2008.

[26] J. Wiehting, M. Bansal, K. Gimpel, and K. Livescu, “Charagram: Embedding Words and Sentences via Character n-grams,” in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, November 2016.

[27] F. Sun, J. Guo, Y. Lan, J. Xu, and X. Cheng, “Inside Out: Two Jointly Predictive Models for Word Representations and Phrase Representations,” in ProceAAAI2016, AAAI Press, Phoenix, AZ USA, February 2016.

[28] M. Rei, G. Crichton, and S. Pyysalo, “Attending to Characters in Neural Sequence Labeling Models,” 2016, https://arxiv.org/abs/1611.04361.

[29] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi, “Sentiment analysis based on improved pre-trained word
embeddings,” *Expert Systems with Applications*, vol. 117, pp. 139–147, 2019.

[30] Z. Rahimi and M. M. Homayounpour, “TensSent: a tensor based sentimental word embedding method,” *Applied Intelligence*, vol. 6, pp. 1–16, 2021.

[31] J. Yu, X. Jian, H. Xin, and Y. Song, “Joint Embeddings of Chinese Words, Characters, and Fine-grained Subcharacter Components,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, September 2017.

[32] H. Peng, Y. Ma, Y. Li, and E. Cambria, “Learning multi-grained aspect target sequence for Chinese sentiment analysis,” *Knowledge-Based Systems*, vol. 148, no. MAY15, pp. 167–176, 2018.

[33] H. Peng, E. Cambria, and X. Zou, “Radical-based hierarchical embeddings for Chinese sentiment analysis at sentence level,” in *Proceedings of the FLAIRS*, pp. 347–352, Palo Alto, CA, USA, May 2017.

[34] O. Wu, T. Yang, M. Li, and M. Li, “Two-level LSTM for sentiment analysis with lexicon embedding and polar flipping,” *IEEE Transactions on Cybernetics*, pp. 1–13, 2020.

[35] J. Wei and K. Zou, “EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks,” 2019, https://arxiv.org/abs/1901.11196.

[36] M. Wenchao, L. Lianchen, and C. Anyan, “A comparative study on Chinese word segmentation using statistical models,” in *Proceedings of the 2010 IEEE International Conference on Software Engineering and Service Sciences*, pp. 482–486, Beijing, China, July 2010.

[37] Github, “Chinese Shopping Reviews sentiment analysis,” 2016, https://github.com/BUPTLdy/Sentiment-Analysis.