Connection Science

ISSN: 0954-0091 (Print) 1360-0494 (Online) Journal homepage: www.tandfonline.com/journals/ccos20

A text classification method based on a convolutional and bidirectional long short-term memory model

Hai Huan, Zelin Guo, Tingting Cai & Zichen He

To cite this article: Hai Huan, Zelin Guo, Tingting Cai & Zichen He (2022) A text classification method based on a convolutional and bidirectional long short-term memory model, Connection Science, 34:1, 2108-2124, DOI: 10.1080/09540091.2022.2098926

To link to this article: https://doi.org/10.1080/09540091.2022.2098926

Published online: 12 Jul 2022.

CONNECTION SCIENCE
2022, VOL. 34, NO. 1, 2108–2124
https://doi.org/10.1080/09540091.2022.2098926

A text classification method based on a convolutional and bidirectional long short-term memory model

Hai Huan[a], Zelin Guo[a], Tingting Cai[b] and Zichen He[b]

[a]School of Electronics & Information Engineering, Nanjing University of Information Science & Technology, Nanjing, People's Republic of China; [b]School of Artificial Intelligence, Nanjing University of Information Science & Technology, Nanjing, People's Republic of China

ABSTRACT

Text classification refers to labelling text with specified labels, and it is widely used in public opinion supervision, spam detection, and other fields. However, due to the complex semantics of natural language and the difficulty of extracting semantic features, users of traditional methods encounter difficulties when trying to achieve better classification results. In response to this problem, a text classification method based on the CBM (Convolutional and Bi-LSTM Model) model, which can extract shallow local semantic features and deep global semantic features, is proposed. First, the text is vectorised using the Glove model in the embedding layer. Then, the vector text is sent to the Multiscale Convolutional Neural Network (MCNN) and the Bidirectional Long Short-Term Memory network (Bi-LSTM) respectively. The Bi-LSTM layer is also designed in the present work with use of mixed attention to extract deeper semantic features. Finally, the MCNN features and Bi-LSTM features are fused and sent to the softmax layer for classification. Experimental results show that the model can significantly improve the accuracy of text classification.

ARTICLE HISTORY
Received 7 March 2022
Accepted 2 July 2022

KEYWORDS
Text classification; feature extraction; multi-head attention mechanism; bidirectional long short-term memory network

1. Introduction

The goal of text classification which is one of the basic tasks of natural language processing, is to assign labels to text. It has a wide range of applications, including sentiment analysis (Qiaoyun et al., 2021; Zhongliang et al., 2022; Shunxiang et al., 2022; Zhao, 2017), question and answer classification (Zhang & Lee, 2003), and topic classification (Cho et al., 2014). Traditional text classification methods use sparse vocabulary features to represent documents, and treat words as the smallest unit, such as support vector machine model, naive Bayes, N-gram model (Kontorovich, 2004), etc. Documents represented by such methods generally exhibit the characteristics of high dimensionality and sparse data, so the classification accuracy is low. Later, with the rise of distributed representation, the use of high-dimensional dense vector representation documents gradually becomes the mainstream, such as the word2vec (Mikolov, 2013) or Glove (Global Vectors for Word Representation)

CONTACT Hai Huan haihuan@nuist.edu.cn School of Artificial Intelligence, Nanjing University of Information Science & Technology, Nanjing 210044, People's Republic of China

(Pennington et al., 2014) models. The word vectors trained by using this type of method represent the contextual semantic information of the text. With the emergence of deep learning in recent years, more researchers use deep learning neural networks for text classification, such as convolutional neural network (CNN) and recurrent neural network (RNN). CNN-based methods all use convolution kernels of the same size to extract features, which can effectively extract local features of text, but also prevents this type of method from being able to extract multi-scale features, and it cannot capture context dependencies between words. RNN-based methods can extract contextual information, but such methods cannot capture key information in the text and are difficult to extract local features in the text. Traditional classification methods are unable to solve more complex classification problems (Jair et al., 2014). To address the shortcomings of CNN and RNN, a CBM model is proposed for complete text classification.

1.1. Main contributions

A CBM model for text classification tasks is proposed. Firstly, a multiscale convolutional neural network (MCNN) module is designed to obtain the shallow local semantic features of the article, making up for the CNN feature extraction. Meanwhile, bidirectional long short-term memory **(**Bi-LSTM) is employed to obtain a global representation of the article to overcome the problem whereby CNN fails to capture the contextual dependencies between words. Secondly, in view of the problem that the Bi-LSTM model cannot capture key information, a MIX attention scheme is designed, and the fusion matrix is adopted to mix each head attention, which can extract key information from different perspectives: the effect thereof is better than that when using a pure attention mechanism. The contributions of this article can be summarised thus:

(1) Proposing a MCNN module. The MCNN can use multiple convolution kernel windows of different sizes to capture shallow features more flexibly, adapt to the diversity of word features, increase the efficiency of the network in extracting shallow features, and improve the accuracy of text classification on multiple data sets.
(2) Proposing the MIX attention module. MIX attention can allow fusion of various header information to capture more comprehensive key information, so that the classification accuracy is further improved.

The rest of this article is organised as follows: Section 2 covers related research methods of existing text classification; Section 3 describes the specific implementation process of the proposed method; Section 4 presents the experimental work; Section 5 concludes.

2. Related work

In 2014, Kim et al. proposed the text convolutional neural network (TextCNN) (Kim, 2014) using CNN as a sentence feature encoder for text classification for the first time. Zhang et al. established character-level convolutional networks (Char-CNN) for text classification in 2015 (Zhang et al., 2015). The network does not consider the intrinsic meaning of words and obtains a more fine-grained representation of text based on a single character; however,

Char-CNN improves the computational complexity of the network and increases the difficulties in practical applications. In 2017, Johnson et al. established the deep pyramid convolutional neural network (DPCNN) for text classification (Johnson & Zhang, 2017). DPCNN uses a pyramid-shaped network architecture to achieve the best accuracy by increasing the network depth (this does not increase computational complexity of the network).

RNN treats the text as a set of word sequences and understands the structure of the text by determining the dependencies between the words to acquire semantic information (Miyamoto & Cho, 2016). The traditional RNN model has the phenomenon of gradient disappearance and gradient explosion. Hochreiter et al. proposed a long short-term memory (LSTM) network (Hochreiter & Jürgen, 1997), which avoided this problem through a set of special gate structures. On this basis, Zhou et al. proposed a Bi-LSTM network combined with two-dimensional maximum pooling to capture text features (Zhou et al., 2016). Bi-LSTM consists of forward LSTM and backward LSTM. The composition can better capture the two-way semantic dependence. Since CNN cannot capture the contextual dependencies between words. To solve this problem, Zhou et al. proposed convolutional LSTM (C-LSTM) (Zhou et al., 2015) in 2015. C-LSTM first uses CNN to extract semantic features at phrase level, and then feeds it to the LSTM to determine the context dependency between words. In 2016, Li et al. proposed a deep stochastic computing convolutional neural network (DSCNN) (Li et al., 2016). DSCNN uses CNN to extract features from the hidden state of LSTM, which captures the context dependency between words to a certain extent. In 2018, Zhao et al. developed a new sandwich structure adaptive learning of local global (ALLG) to learn local semantic representation and global structure representation (Zhao et al., 2018) and proposed two strategies to cope with the feature fusion problem.

The attention mechanism (Vaswani & Shazeer, 2017) can describe the dependency of the context and capture the key information. In 2019, Chia proposed Transformer to CNN (Trans-CNN) method (Chia et al., 2019). Trans-CNN is a hybrid model based on CNN and a self-attention mechanism, which is trained through the distillation process of a large-scale pre-training model, that is, using a large-scale "teacher model". The language model trains a small-scale "student model" structure, which reduces the model scale and computational cost. In 2020, Gu et al. proposed a method based on the attention mechanism (Gu & Peng, 2020), which uses a convolution operation to extract attention signals, highlighting the emotional words and turning words of the focus of the text. Li et al. established the LSTM_CNN Hybrid model (Li & Ning, 2020), which first uses LSTM to learn the long-term dependence of the text, and then designs a shallow convolution structure to extract the semantic features of the text, and finally uses the maximum pooling operation to filter useful and important features for classification. In 2021, Deng et al. constructed an attention-based gating mechanism network (Deng et al., 2021), citing the gating mechanism to assign weights to Bi-LSTM and CNN output features to acquire text fusion features that are conducive to classification. In the same year, Tam et al. proposed the convolutional bidirectional long short-term memory (ConvBiLSTM) deep learning model (Tam & Said, 2021), which integrates CNN and Bi-LSTM to realise sentiment analysis. In recent years, due to the rise of large-scale pre-training models, Sun et al. proposed an enhanced representation through knowledge integration (ERNIE) pre-training model based on knowledge enhancement (Sun et al., 2019). The ERNIE model predicts semantic units such as words and entities so that the model can learn the semantic representation of complete concepts. On this basis, Cheng et al. established the bidirectional gate recurrent unit (ERNIE_ BiGRU)

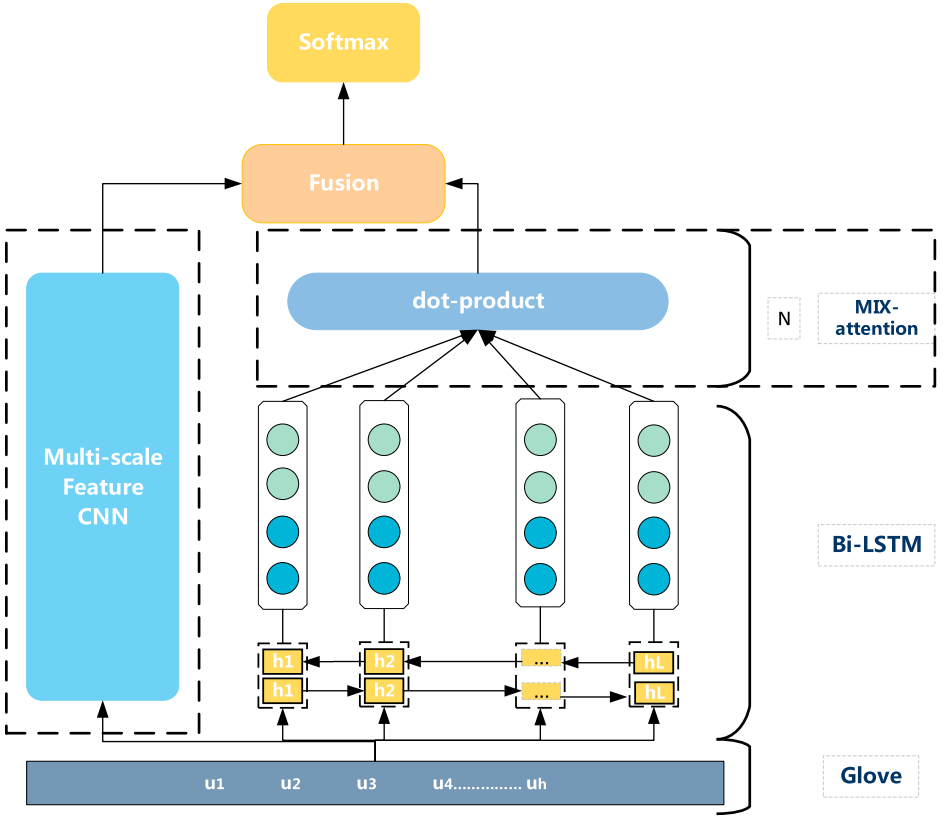

Figure 1. CBM network.

model (Cheng et al., 2021). This model uses ERNIE to model the prior semantic knowledge unit directly, which enhances the semantic representation ability of the model, and then uses BiGRU to capture the contextual dependencies between words. Inspired by hierarchical attention networks (Yang et al., 2016), Deng and Ren established a hierarchical emotion recognition model based on label embedding (Deng & Ren, 2021). The model trains the label embedding matrix through joint learning to learn the contextual information while determining the emotional representation of the sentence.

3. Method

The overall structure of the CBM network is illustrated in Figure 1. The dotted line part represents the main contribution of the present work. The bottom of the figure is the Glove word vector, the left half is the MCNN, and the right half is Bi-LSTM and MIX attention. First, the model loads the Glove word vector to vectorise the original corpus. Then, the parallel structure of the MCNN and Bi-LSTM is designed to extract text features. The MCNN module captures the local features of the text; the Bi-LSTM module is employed to extract the global information of the article. MIX attention is adopted to capture the key information in the global information, and further improve the weight of key features in text classification; finally, the two are combined and the softmax function is used for classification.

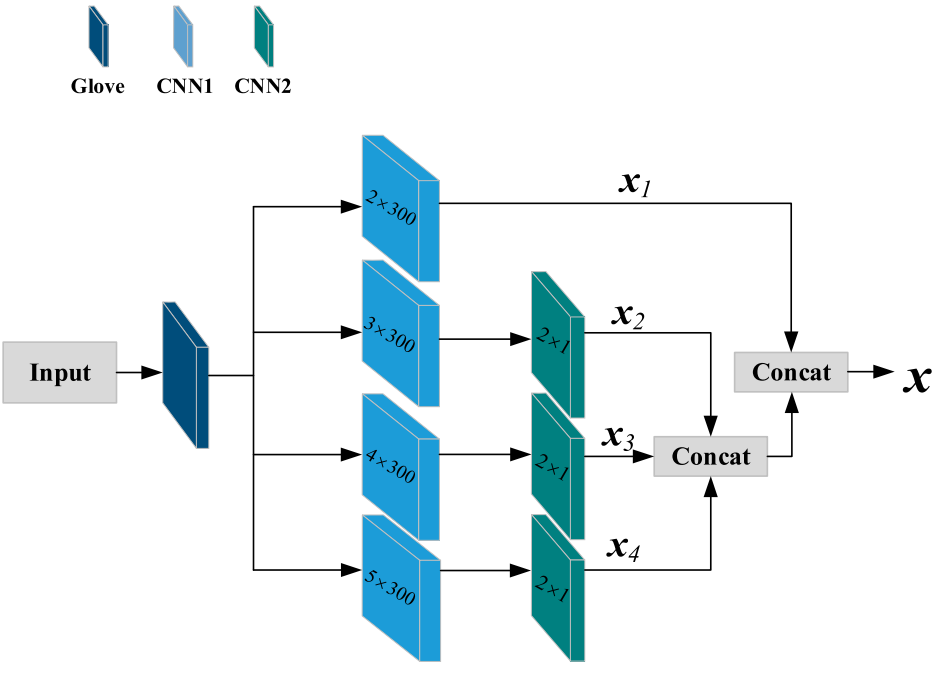

Figure 2. Internal structure of the MCNN.

3.1. Shallow feature extraction module

3.1.1. Word vector model Glove

The Glove method is employed to extract and express language features [6]. It is a word characterisation tool based on global word frequency statistics, which can express a word as a vector composed of real numbers. Glove can calculate the semantic similarity between two words by operating on the vector. Compared with word2vec, it considers contextual information in the global scope and uses the co-occurrence matrix to introduce global information; compared to the previous method, the Glove model introduces a weight function to control the relative weight of words.

3.1.2. Multiscale CNN

CNN has good position invariance (Kim, 2014), so the model uses the convolution kernel window to slide on the document representation matrix, which can extract the phrase-level features of the document and use it for the text to determine the local feature representation thereof (Jincheng & Wang, 2022; Suruchi, 2021). The internal structure of the MCNN is displayed in Figure 2. The input in the figure is the vector text that has gone through the Glove model. The module consists of four parallel channels. Channel 1 uses a convolution kernel with a window size of 2×300 to undertake convolution operations on the input signal to extract local features x_1. Channels 2–4 use convolution kernels with window sizes of 3×300, 4×300, and 5×300 to conduct convolution operations on the input signal. The signal undergoes a convolution operation, then a convolution kernel with a window size of 2×1 is used to perform a convolution operation on the convolved signals from Channel 2 to Channel 4 again to further extract local features x_2, x_3, x_4. This design enables the text

information to be gathered after initial feature extraction, so that the network can learn both "sparse" (3×300, 4×300, and 5×300) local features and "non-sparse" (2×1) local features. That is, features of different scales are extracted through receptive fields of different sizes. Finally, the CONCAT operation is adopted to fuse the local features extracted from the four channels to obtain the final shallow local feature representation x.

If the sentence has h words, $u_i \in R^{300}$ represents the 300-dimensional vector representation of the ith word in the sentence, and $u_{i:i+h-1}$ is the concatenation of h words from the word u_i to the word u_{i+h-1}. The convolution operation on the window from the word u_i to u_{i+h-1} is adopted to generate the feature vector x_i. The specific convolution calculation formula is as follows:

$$\begin{aligned} x_i &= Conv(u_i, u_i + 1, \ldots, u_i + h - 1) \\ &= relu(W \bullet u_i : i + h - 1 + b) \end{aligned}$$

In formula (1), $W \in R^{h \times 300}$ denotes the parameter matrix, $b \in R$ is the bias term, and *relu* represents the activation function. The convolution operation is applied to the entire text representation matrix to obtain the feature x_i, and then all obtained features x_i are spliced to produce a feature map $x \in R^{n-h+1}$:

$$x = [x1 : x2 : x3 : \ldots ; xn - h + 1] \tag{2}$$

3.2. Deep feature extraction module

3.2.1. Bi-LSTM network

The Bi-LSTM is used (Zhou et al., 2016), which overcomes the problems of gradient disappearance and gradient explosion that are easy to appear in RNNs through a set of gate structures. A single LSTM unit contains three gates (Hochreiter & Jürgen, 1997): a forget gate $f^{(t)}$, an input gate $i^{(t)}$, and an output gate $o^{(t)}$. Its internal structure is illustrated in Figure 3.

Among them, $c^{(t)}$ represents the storage state unit at time t; $h^{(t)}$ is the output produced by the LSTM at time t, and $x^{(t)}$ denotes the input of the model at time t. The forget gate determines how much the cell state $c^{(t-1)}$ at the previous moment is retained to the current state $c^{(t)}$, the input gate determines how much of the network input $x^{(t)}$ at the current moment is saved to the cell state $c^{(t)}$, and the output gate how much of the control unit state $c^{(t)}$ is output to the current output value $h^{(t)}$ of the LSTM. The formulae for input gate $i^{(t)}$, forget gate $f^{(t)}$, and output gate $o^{(t)}$ are expressed as follows:

$$i^{(t)} = \sigma(W_{xi}x^{(t)} + U_{hi}h^{(t-1)} + b_i) \tag{3}$$

$$f^{(t)} = \sigma(W_{xf}x^{(t)} + U_{hf}h^{(t-1)} + b_f) \tag{4}$$

$$o^{(t)} = \sigma(W_{xo}x^{(t)} + U_{ho}h^{(t-1)} + b_o) \tag{5}$$

The current state $c^{(t)}$ is updated by $g^{(t)}$, and finally $o^{(t)}$ and $c^{(t)}$ are employed to calculate the current output value $h^{(t)}$. The calculation formulae are as follows:

$$g^{(t)} = \tanh(W_{xg}x^{(t)} + U_{hg}h^{(t-1)} + b_g) \tag{6}$$

$$c^{(t)} = f^{(t)} \odot c^{(t-1)} + i^{(t)} \odot g^{(t)} \tag{7}$$

$$h^{(t)} = \tanh(c^{(t)}) \odot o(t) \tag{8}$$

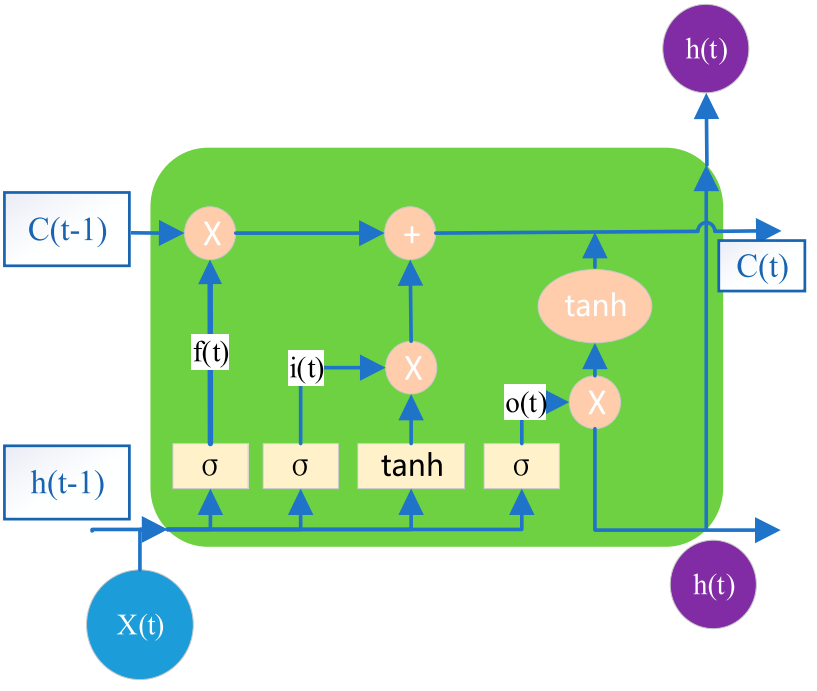

Figure 3. Single LSTM internal unit.

Among them, b_g, b_f, b_i and b_o are the biases of each gate, and U_{hi}, U_{hf}, U_{ho} and U_{hg} are cycle weights. The symbol $\odot$ represents the Hadamard product operator, denoting element-wise multiplication; tanh refers to the hyperbolic tangent function, and σ is the sigmoid activation function.

The internal structure of Bi-LSTM is shown in Figure 4. There are two LSTM units in each module, which are adopted to calculate the forward hidden sequence $\vec{h}$ and the backward hidden sequence $\overleftarrow{h}$. Through iterations, the forward hidden layer state from time $t = 1$ to $t = T$ and from time $t = T$ is spliced to the backward hidden layer state with $t = 1$ to produce output sequence h. As shown in formula (9), the state information of a single hidden layer is:

$$h = [\vec{h} \oplus \overleftarrow{h}] \tag{9}$$

$\oplus$ is the concatenation operator, that is, addition of element-by-element.

3.2.2. MIX attention scheme

Since the attention mechanism can capture the key information in the sentence (Vaswani & Shazeer, 2017), it can solve the problem that Bi-LSTM cannot extract the key features in the article. This article further proposes a MIX attention scheme on this basis, and its internal structure is described as follows.

Figure 5 shows MIX attention. MIX attention is composed of multiple dot-products. The text data are fed into the MIX attention module after passing through the Bi-LSTM model. In MIX attention, V represents the value matrix with dimension d_v, K is the key matrix with dimension d_k, Q denotes the query matrix with dimension d_k, V, K and Q are all determined

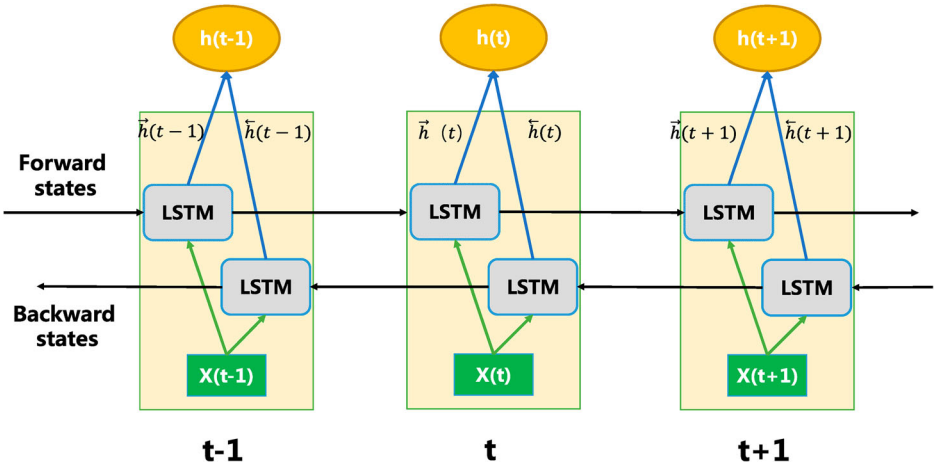

Figure 4. Bi-LSTM internal structure diagram.

by multiplying the input by the matrix with dimensions d_v, d_k and d_k. The previous attention mechanism (Vaswani & Shazeer, 2017) uses the transposition of the query matrix and all the key matrices as a dot product and employs the softmax function to determine the weights:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{10}$$

This kind of calculation method separates each head attention, so that each attention is isolated and has no connection with any other, which affects the extraction efficiency of key features. In order to further improve the feature extraction ability of the attention mechanism and improve the accuracy of text classification, the MIX module is designed in a dot-product form (Figure 6).

Figure 6 shows the internal structure of the MIX attention module. First, the Q matrix and K matrix of each head are transposed before application of the dot product operation, and we then divide by $\sqrt{d_k}$ by way of scaling to get the information of each head $\hat{J}^{(i)}$, where i is the number of the attention head. The specific calculation formula is as follows:

$$\hat{J}^{(1)} = \frac{Q^{(1)}K^{(1)^T}}{\sqrt{d_k}}, \hat{J}^{(2)} = \frac{Q^{(2)}K^{(2)^T}}{\sqrt{d_k}}, \hat{J}^{(3)} = \frac{Q^{(3)}K^{(3)^T}}{\sqrt{d_k}} \ldots \hat{J}^{(h)} = \frac{Q^{(h)}K^{(h)^T}}{\sqrt{d_k}} \tag{11}$$

Among them, h is the number of attention heads. The original calculation method directly performs a softmax operation on the head information and then multiplies it by the V matrix, which leads to the inability of each head information to interact and reduces the feature extraction efficiency of the attention mechanism. As shown in Figure 6, these headers are multiplied by a randomly initialised fusion matrix, and the feature $J^{(i)}$ fused with each header information is obtained, so that $J^{(i)}$ captured by each head can be used in an interactive manner. Thereafter, the softmax function is used to normalise the weight of the

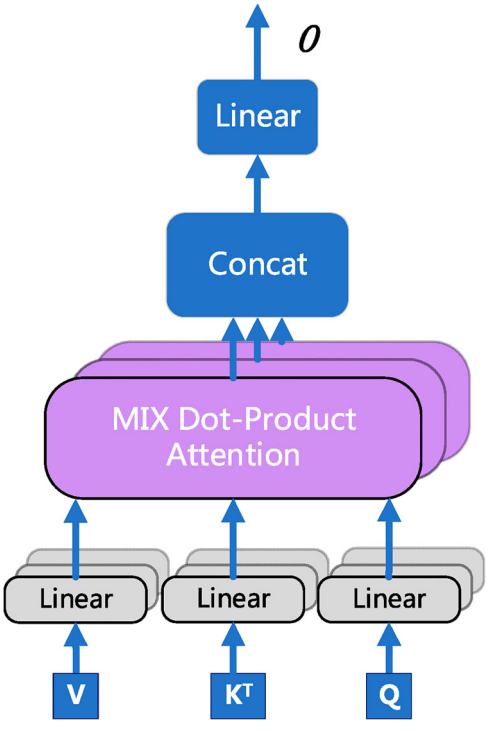

Figure 5. MIX attention.

feature $J^{(i)}$ to obtain $P^{(i)}$, as follows:

$$\begin{pmatrix} J^{(1)} \\ J^{(2)} \\ J^{(3)} \\ . \\ . \\ . \\ J^{(h)} \end{pmatrix} = \begin{pmatrix} \lambda_{11} & \lambda_{12} & \lambda_{13} & \dots & \lambda_{1h} \\ \lambda_{21} & \lambda_{22} & \lambda_{23} & \dots & \lambda_{2h} \\ \lambda_{31} & \lambda_{32} & \lambda_{33} & \dots & \lambda_{3h} \\ . & . & . & & . \\ . & . & . & . & . \\ . & . & . & & . \\ \lambda_{h1} & \lambda_{h2} & \lambda_{h3} & \dots & \lambda_{hh} \end{pmatrix} \begin{pmatrix} \hat{J}^{(1)} \\ \hat{J}^{(2)} \\ \hat{J}^{(3)} \\ . \\ . \\ . \\ \hat{J}^{(h)} \end{pmatrix} \tag{12}$$

$$P^{(1)} = \text{softmax}(J^{(1)}), P^{(2)} = \text{softmax}(J^{(2)}) \dots\dots P^{(h)} = \text{softmax}(J^{(h)}) \tag{13}$$

$P^{(i)}$ is an intermediate variable fused with other attention head information, finally, $P^{(i)}$ is multiplied by the respective V matrices, and each head attention is concatenated to obtain the final representation O of the deep global semantic features as follows:

$$O^{(1)} = P^{(1)}V^{(1)}, O^{(2)} = P^{(2)}V^{(2)} \dots\dots O^{(h)} = P^{(h)}V^{(h)} \tag{14}$$

$$O = [O^{(1)}, O^{(2)}, O^{(3)} \dots\dots O^{(h)}] \tag{15}$$

The network obtains the global features of the text through Bi-LSTM, and then passes the output of Bi-LSTM to MIX attention, so that the network can extract the key features of the text again, thus forming a deep feature extraction module.

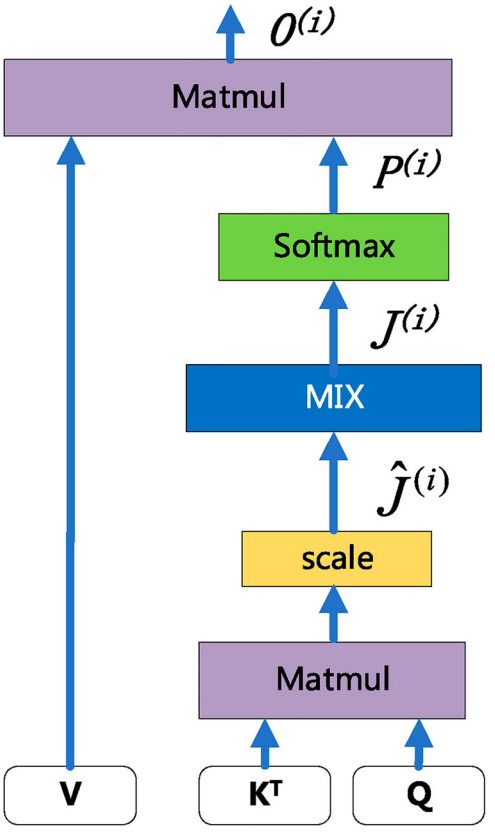

Figure 6. MIX attention internal structure dot-product.

Table 1. Data set information (SA stands for sentiment analysis, QA stands for question and answer).

	AGnews	DBPedia	text_class	Yahoo!Answer	Yelp.P	Amazon.F
Task type	News	Ontology	News	QA	SA	SA
Train dataset	120k	560k	8k	1.4M	560k	3.6M
Test dataset	7.6k	70k	1k	60k	38k	400k

Finally, in this model, the two features extracted by the MCNN and Bi-LSTM are fused through the fusion module. The splicing function is used as the fusion method of the two, that is, aligning the tensor dimensions of *x* and *O* above and using the addition operation for fusion.

4. Experiment

4.1. Data set introduction

Six text classification data sets are used (Table 1). All six datasets can be downloaded from https://paperswithcode.com/datasets AGnews is a four-category news corpus, and DBPedia is a 14-category ontology data set from Wikipedia (Zhilin et al., 2019). Yelp.P is a review corpus that predicts sentiment categories and is a dual-classification data set. Yahoo!

Answers is a question and answer data set, with a total of 10 categories (Zhang et al., 2015). text_class is a 20-category English data set. Amazon.F is an Amazon review data set with five categories (Zhilin et al., 2019).

4.2. Experimental details

The 300-dimensional Glove word vector (Pennington et al., 2014) is used for word embedding. When training the model, the word embeddings are updated along with other parameters. Stochastic gradient descent (SGD) is used as an optimiser for all trainable parameters. The experiment is run on a high-performance computer equipped with an NVIDIA T1 graphics card and 32-GB RAM using a pytorch 0.4.1 framework and Python 3.6.

To achieve better classification effect, some parameter tuning techniques are also adopted. First, a good experimental environment is configured, and the settings of all experimental variables for ease of adjustment can be centralised. Second, referring to the experimental parameters in the classic papers, the batch size, learning rate, etc are adjusted. Because an excessively batch size usually leads to insufficient video memory, in the experiment, the batch number is adjusted from large to small. When setting the learning rate, the situation of different data sets varies. Herein, we use an order of 10, and generally choose 0.01 or 0.001. In the MCNN module, the dimension of each output channel is set to 256; inside Bi-LSTM, the hidden layer unit is also set to 256. In MIX attention, the number of dot-products is set to 8, and the dimensions of V, K, and Q are all set to 64.

To prevent overfitting, the following measures are also taken: first of all, the dropout strategy is adopted in the training process, that is, a certain proportion of hidden layer units in the dropout layer (usually about 0.5) is randomly discarded. Since the hidden layer units discarded in each iteration are different, the network can return correct classification results through some neural units after multiple iterations, avoiding overfitting. Secondly, we also use an early stopping strategy, that is, stopping iterations before the model converges on the training dataset iterations to prevent overfitting. The specific approach is to calculate the accuracy of the validation set at the end of each epoch and stop training when the accuracy no longer improves over multiple consecutive epochs.

4.3. Experimental analysis

Glove is used as the text representation method and conducts improvement experiments on three English data sets. Model 1 is used as the baseline model with a single model CNN for classification. Model 2 uses the MCNN proposed herein for classification. Model 3 is a two-way long and short-term network plus attention mechanism. Model 4 integrates the proposed MCNN with Bi-LSTM. Model 5 is a hybrid model MCNN_Bi-LSTM_attention, which uses the MCNN and Bi-LSTM fusion attention mechanisms. Model 6 is the CBM model proposed herein: six groups of comparative experiments are conducted, and the experimental results are displayed in Table 2.

The experimental results indicate that the proposed model has achieved the best classification effect on all data sets. On the 20-category English data set, the accuracy reaches 82.24%, on the four-category data set AGnews, the accuracy reaches 92.50%, and on the 14-category DBPedia data set, the accuracy reaches 98.80%. Comparing Model 6 (proposed herein) with other single models and hybrid models shows that: compared with the

Table 2. Classification improvement experiments (%).

Module	text_class	AGnews	DBPedia
1. Glove + CNN	79.50	91.88	98.42
2. Glove + MCNN	79.78	92.23	98.54
3. Glove + Bi-LSTM _attention	76.30	90.28	97.37
4. Glove + MCNN_ Bi-LSTM	79.63	92.05	98.65
5. Glove + MCNN_ Bi-LSTM _attention	80.13	92.11	98.35
6. Glove + MCNN_ Bi-LSTM _MIXattention (ours)	**82.24**	**92.50**	**98.80**

baseline model CNN, the proposed model is improved by 2.74%, 0.62%, and 0.38%, respectively. The performance on the three data sets is better than the baseline model CNN; in particular, on the text_class data set, the improvement in classification accuracy is significant. Comparing Model 2 with Model 1, Model 5 with Model 3, and under the same control of other modules, the difference between these two groups of experiments is whether to use the MCNN module. From the data in Table 2, it can be found that in terms of classification accuracy, Model 2 and Model 5 using the MCNN module are better than Model 1 and Model 3. The improvement of the network performance of the MCNN module also verifies the effectiveness of the module. Models 4–6 are mixed models, and their accuracy on each data set has been improved compared to that achieved with a single model. The hybrid model can extract local features and global semantic features at the same time, making it more comprehensive than the single model, thus affecting the classification effect thereof. This result indicates that extracting shallow local features through the MCNN can significantly improve network performance, and the comparison with a single model also proves that the effect of the hybrid model is better than that of the single model.

Comparing Model 6 with Model 4, it can be seen from the table that there are 2.61%, 0.45%, and 0.15% improvements on the three data sets, respectively. Both are hybrid models using the MCNN and Bi-LSTM. Model 6 adopts the MIX attention module designed in the present research; the proposed model is then compared with Model 5. In the case of the same use of the MCNN and Bi-LSTM, the only difference is the attention mechanism and MIX attention. From the tabulated data, Model 6 (proposed herein) that is used on the text_class data set is 1.11% more accurate than Model 5; on AGnews, the accuracy is increased by 0.39%; on DBPedia, the accuracy is increased by 0.45%. Although the attention mechanism has achieved excellent classification results, the accuracy has reached more than 90% on both data sets, but the proposed MIX attention scheme is still improved on this basis, proving the effectiveness of the module. The MIX attention module mixes each head of information, and the extracted features are richer and more detailed than the pure attention mechanism, so the classification accuracy is also higher. The result also shows that the MIX attention scheme designed in the present research is better than the original attention mechanism.

To further prove the superiority of the proposed model, the outputs are compared with several other groups of models that also use CNN and Bi-LSTM in joint learning (Liu & Guo, 2019; Pradhan et al., 2021; Song, 2018; Trueman & Cambria, 2021), see Table 3 for details.

It can be seen from Table 3 that the proposed model is based on the same method on the three datasets. Models 1–4 are all series structures in terms of structure, that is, the Bi-LSTM network is followed by CNN. However, the proposed model is a parallel structure. It is believed that there is position information of the sequence in the text. If the method of

Table 3. Accuracy of text classification: metric: classification accuracy rate (ACC, %).

Module	text_class	AGnews	DBPedia
1. CNN + Bi-LSTM (Song, 2018)	75.67	89.88	98.59
2. CNN + Bi-LSTM + attention (Liu & Guo, 2019)	72.59	91.67	98.57
3. CNN + LSTM_Bi − LSTM + attention (Pradhan et al., 2021)	70.44	90.40	98.24
4. CNN + Bi-LSTM_Bi − LSTM + attention (Trueman & Cambria, 2021)	70.21	91.60	98.44
5. MCNN_Bi-LSTM_MIXattention (ours)	**82.24**	**92.50**	**98.80**

Table 4. Accuracy of text classification: metric: classification accuracy rate (ACC, %).

Module	AGnews	DBPedia	Yelp P	Yahoo.Answer	Amazon
Capsnets (Ren & Lu, 2018)	92.40	98.30	**96.50**	–	61.00
Char-CNN (Zhang et al., 2015)	90.49	98.45	95.12	71.20	59.57
Char-CRNN (Xiao & Cho, 2016)	91.40	98.60	94.50	71.40	59.20
VDCNN (Alexis & Holger, 2016)	91.30	98.70	94.70	71.75	**63.00**
ALLG (Zhao et al., 2018)	90.55	–	95.20	72.16	**63.00**
Trans-CNN (Chia et al., 2019)	91.20	98.50	–	71.00	–
Bag of tricks (Joulin, 2017)	91.50	98.10	93.80	72.00	55.80
CNN_Bi-LSTM_MIX attention (ours)	**92.50**	**98.80**	93.56	**72.24**	58.49

The bolded values in the table are the results of this model.

the serial structure is used, the position information in the text sequence cannot be captured by the Bi-LSTM network, which will cause certain feature loss. The proposed method can avoid this problem. In addition, the MCNN module and MIX attention proposed herein can enhance the feature extraction ability of the network and are also better than similar methods in terms of classification accuracy.

To verify the reliability of the model, another experiment is conducted on the other five data sets and the result is compared with the classic model (Table 4): the data pertain to neural network models using different methods, such as capsule network (Ren & Lu, 2018); CNN-based models include: Char-CNN (Zhang et al., 2015), very deep convolutional neural network (VDCNN) (Alexis & Holger, 2016), topic attention networks for neural topic modelling (TAN-NTM) (Panwar, 2020), Trans-CNN (Chia et al., 2019), and Char-CRNN (Xiao & Cho, 2016).

It can be seen from Table 4 that the proposed feature-fusion method outperforms other methods on three of the data sets, being 1–2% higher than the model based on CNN or RNN. On the 4-category AGnews news data set, the accuracy is up to 92.50%; on the 14-category ontology data set DBPedia, the accuracy is up to 98.80%; on the 10 categories of Yahoo! on the Answers question and answer data set, an accuracy of 72.24% is achieved, which is higher than other models. Compared with a single model, such as Capsnets, Char-CNN, etc., the accuracy of the proposed model is increased by 0.26–2.01%. The experimental results imply that the model is better than the single model, and it also proves that the mixed model is better than the single model in text classification, which is in agreement with the conclusion drawn above. For the Yelp P and amazon datasets, optimisation is not achieved: these two data sets contain longer sequences and variable grammatical information; secondly, those model structures are more suitable for this type of data set, because this type of data set has certain advantages; in the end, those authors used some special training techniques during training, which can improve the classification performance of the model. Compared with the hybrid model Trans-CNN, the accuracy of the proposed

Table 5. Accuracy of text classification: metric: classification accuracy rate (ACC, %).

Dataset	Model	Accuracy
AGnews	ULRIDTC (Chu et al., 2020)	85.20
	LCLETC (Guo et al., 2020)	90.42
	AFTB (Ghazi et al., 2021)	91.66
	Ours	**92.50**
DBpedia	ULRIDTC (Chu et al., 2020)	97.40
	LCLETC (Guo et al., 2020)	98.34
	AFTB (Ghazi et al., 2021)	98.31
	Ours	**98.80**
Yelp P	SSVAE (Ghazi et al., 2021)	92.85
	TAN-NTM (Panwar, 2020)	88.90
	Ours	**93.56**
Yahoo.Answer	ULRIDTC (Chu et al., 2020)	66.70
	Ours	**72.24**

model is 0.21–1.24% higher than on each data set, indicating that the hybrid method is better than Trans-CNN.

To verify the effectiveness of the model, the latest text classification methods are collected and compared (Table 5); these methods include ULRIDTC (Unsupervised Label Refinement Improves Dataless Text Classification) (Chu et al., 2020) based on label enhancement, LCLETC (Label Confusion Learning to Enhance Text Classification) (Guo et al., 2020), AFTB (Adversarial Fine-Tuning BERT) (Javid et al., 2021) based on adversarial learning and SVSAE (Semi-Supervised VAE) based on semi-supervised learning (Ghazi et al., 2021). On the AGnews data set, the proposed model achieves a classification accuracy of 92.5%, which is higher than the other three methods; on the DBpedia data set, the proposed model achieves a classification accuracy of 98.8%, which is 1.40% higher than the classification accuracy of ULRIDTC. On the Yelp P data set, the proposed method achieves an accuracy of 93.56%, which is 0.71% higher than SVSAE and 4.66% higher than TAN-NTM. On the Yahoo Answers dataset, the proposed model achieves a classification accuracy of 72.24%, which is 5.54% higher than ULRIDTC.

In summary, the method proposed herein can improve the effect of text classification, and achieve the best results compared with other new method developed in the last two years. The text classification model fused with shallow features and deep features proposed herein can significantly improve the accuracy of text classification.

5. Conclusion

Text classification is an important task in the field of natural language processing, and it is widely used in tasks such as topic classification and public opinion analysis. Classification using CNN or RNN is a classic method in this field. However, they all show their own limitations: the performance of CNN is limited by the size of the convolution kernel window and cannot capture the context dependence between words; RNN cannot effectively extract the key features of the sentence. To solve these problems, based on the CBM model, an MCNN module is designed to extract the local features of the text, and the MIX attention is designed to extract the key features of the sentence. The effectiveness of the proposed module is verified through comparative experiments. On the three benchmark