applied sciences

MDPI

*Article*

# SA-SGRU: Combining Improved Self-Attention and Skip-GRU for Text Classification

**Yuan Huang, Xiaohong Dai *, Junhao Yu and Zheng Huang**

School of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China
* Correspondence: dxhong163@163.com

**Abstract:** When reading texts for text classification tasks, a large number of words are irrelevant, and in text classification tasks, the traditional self-attention mechanism has the problem of weight distribution limitations. Therefore, a text classification model that combines an improved self-attention mechanism with a Skip-GRU (Skip-grate recurrent unit) network (SA-SGRU) is proposed in this paper. Firstly, Skip-GRU, the enhanced model of GRU (Grate Recurrent Unit), is used to skip the content that is not important for text classification when reading texts and only capture effective global information. Then, the improved self-attention mechanism is introduced to redistribute the weight of the deep text sequences. Secondly, the optimized CNN (convolutional neural network) is combined to bring up the local features of texts. Finally, a Softmax classifier is used to obtain the classification results of sample labels. Experimental results show that the proposed method can achieve better performance on three public datasets compared with other baseline methods. The ablation experiments also demonstrate the effectiveness of each module in the proposed model.

**Keywords:** text classification; skip gated recurrent unit; self-attention mechanism; convolutional neural network



**Citation:** Huang, Y.; Dai, X.; Yu, J.; Huang, Z. SA-SGRU: Combining Improved Self-Attention and Skip-GRU for Text Classification. *Appl. Sci.* **2023**, *13*, 1296. https://doi.org/10.3390/app13031296

Academic Editor: Valentino Santucci

Received: 22 December 2022
Revised: 11 January 2023
Accepted: 14 January 2023
Published: 18 January 2023



## 1. Introduction

With the continuous development of the Internet, Internet text data is growing exponentially. Now it is an urgent problem to apply text classification technology to classify a huge number of diverse pairs of texts scientifically and effectively. As one of the most efficient natural language processing technologies, text classification has been widely used in the fields of public opinion analysis [1], text content retrieval [2], sentiment analysis of online comments [3], etc. While relevant research has attracted attention, more and more people are looking for ways to make insight discoveries and improve decision-making by utilizing the ability provided by machine learning big data. Big data can be defined as "huge data sets beyond the storage, capture, management, or analysis capabilities of traditional database software" [4]. The potential of this data depends on the ability to extract value from vast amounts of data through data analytics. Testing big data is challenging because of its diversity and complexity [5].

Deep learning is a subfield of machine learning that enables performance improvements through data insight [6]. Machine learning algorithms are rarely challenged by big data in acquiring knowledge. Big data provides a large amount of data and information, which can be used by machine learning algorithms to extract patterns or build analysis models [7]. Methods based on deep neural networks, like the convolutional neural network (CNN) and the recurrent neural network (RNN), can extract deep-level features of text and have been widely known and used. The main task of the text classification method based on deep learning is to extract the features of the text and categorize the extracted text features through the neural network.

In recent years, with the in-depth study of text classification algorithms by researchers, it has been proven that deep learning methods are superior to traditional machine learning

methods and have achieved good results in the field of natural language processing. However, these algorithms have some defects in the field of text classification, such as CNN's inability to tackle the long-distance dependence problem in the text information, and RNN always reads the entire text input, resulting in a lengthy input process for longer sequences and insensitivity to key pattern information in the text. Therefore, it is imperative to solve the problems inherent in the above methods.

In this paper, a method called SA-SGRU is presented to address the shortcomings of the aforementioned methods. The SA-SGRU improves GRU and self-attention, making up for their shortcomings while effectively utilizing their advantages. In addition, the optimized CNN is effectively combined to extract richer text features and improve the performance of the model text classification.

The rest of this paper is organized as follows: Section 2 introduces related work. Section 3 presents the proposed method. Section 4 describes the performance of the experiment using the proposed method. Section 5 discusses our contributions and shortcomings, as well as future work. Section 6 summarizes our research.

## 2. Related Works

The ever-increasing importance of structured data in different applications [8] is no exception in text classification. Before using neural networks for text classification, the text needs to be transformed into a structured feature-semantic representation. Fan et al. [9] combined the BERT and CNN algorithms to classify news texts. The core idea is to send BERT to CNN as an embedded layer. This method is superior to the simple BERT and CNN models. Zeng et al. [10] proposed a classification method based on statement hierarchy and CNN. This method not only combines the improved TF-IDF algorithm and Word2vec technology but also uses CNN to extract text features and realize text classification. Lu [11] used BERT and LSTM to build a keyword classification model for journal papers, and the results were significantly improved compared with traditional methods. With a deep and narrow neural network, BERT, a novel linguistic expression mode proposed by the Google team [12], combined context in all neural network layers during training to achieve more accurate text prediction.

Deep learning has revolutionized computer vision, natural language understanding, speech recognition, information retrieval, and more [13]. As one of the most common tasks in natural language processing, text classification is also deeply influenced by deep learning. The text deep network model based on hybrid neural network composition can extract more comprehensive, richer, and higher-level semantic information, which outperforms single networks like CNN and RNN for text classification and is the focus of recent research [14]. Deng et al. [15] proposed a new text classification model called attention-based BiLSTM that fused CNN with a gating mechanism (ABLG-CNN). The attention mechanism is used to derive keyword information, and BiLSTM captures context features. Based on this, CNN captures topic salient features, and a gating mechanism is introduced to assign weights to BiLSTM and CNN output features to obtain text fusion features that are favorable for classification, which achieved good results compared with the traditional models. Kejia Chen et al. [16] proposed an optimized multichannel CNN combined with BiGRU, in which the global information extracted by BiGRU is stitched with the local features extracted by multichannel CNN and passed into the text classifier, thereby addressing the problem that a single neural network cannot obtain global semantic information and the gradient of a traditional RNN disappears. Chen Qian et al. [17] proposed a multi-label classification of options based on the hybrid attention Seq2Seq model, where a multiheaded attention mechanism is used, replicating multiple warheads but with different weight coefficients because the initialization is different, which can better assign weights to the text. Jiming Hu et al. [18] classified policy texts using the CNN-BILSTM-Attention model, which fully utilized the semantic features of policy texts and produced more accurate and efficient classification results than those previously models. Feng et al. [19] proposed a multi-graph attention mechanism model (MCNN-MA) based on

multi-channel convolutional neural networks. In this model, word features, part of speech features, location features, and dependency syntactic features were combined to form three new combination features, respectively, so as to better extract text features and achieve the best performance of the model. Li et al. [20] proposed a combined model of the self-attention mechanism and BiLSTM for document-level text classification tasks, where the global semantic information of text is extracted by bi-directional LSTM and the representation of text features is improved by the self-attention mechanism. Iyad Alagha [21] proposed a method based on knowledge-based features with multilevel attention mechanisms for short Arabic text classification, and the attention mechanism used in this method functions as a category filter, highlighting the most important features while reducing the influence of inappropriate features and improving the classification results.

Du et al. [22] propose a novel knowledge-based Leap-LSTM framework, the core idea of which is to partially supervise the word skipping process through the manual or semi-automatic creation of in-domain keywords and other lexical resources. The jump network improves the efficiency of the model's operation and gets more accurate prediction results. Krzystof et al. [23] propose a text truncation method called Text Guide. The approach reduces the original text length to a predefined limit, which not only improves the performance of naive and semi-naive methods, but also keeps the computational cost low.

## 3. The Proposed Method

Figure 1 describes the text classification model SA-SGRU proposed in this paper. The whole framework consists of three parts: (1) Skip-GRU and an improved self-attention mechanism module; (2) an optimized CNN based on a multi-size convolutional kernel module; and (3) a reordering module.

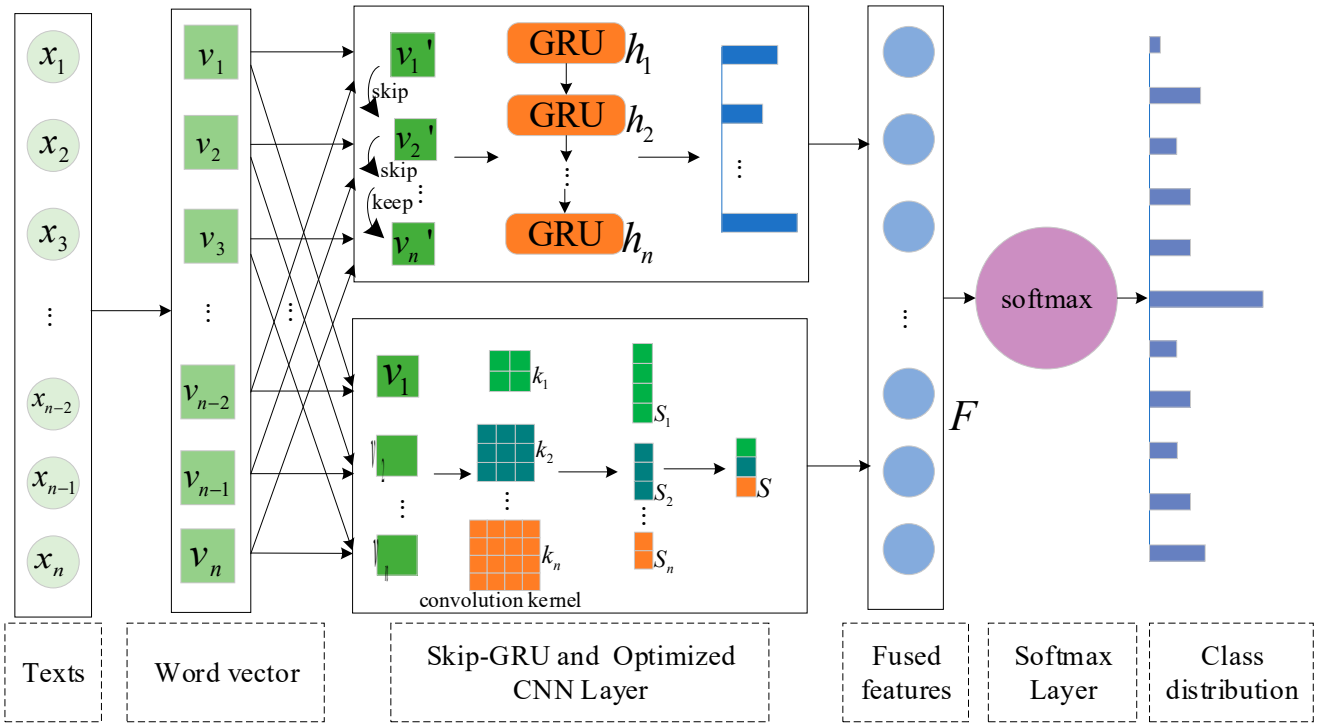


**Figure 1.** Overall structure of SA-SGRU.

An example illustrating the proposed framework of SA-SGRU: suppose that the input text sequence is "This is a very sad story," denoted by the vector $V = v_1, v_2, v_3, v_4, v_5, v_6$ corresponding to the input text sequence obtained through the embedding layer. First, module (1) is used to capture global effective information on the input text, $V$ which

first passes through the Skip network to obtain the vector $V' = v_4', v_5', v_6'$ for the more important content in the classification result (assuming that the three words " this," "is," and "a" in the sequence are skipped) and is then input $V'$ into the GRU network. The feature vector $H = h_1, h_2, h_3$ with global information is obtained, and finally, the key feature information $H'$ is obtained by redistributing the weight information through the improved self-attention mechanism. Second, in order to make up for the lack of sensitivity of module (1) to the local key information, module (2) is used to extract the local key information of the text with multi-size convolutional kernels. Feature vector $V$ obtained after the convolution operation of different sizes by $k_1, k_2, k_3$ (the value of $k_1, k_2, k_3$ in this paper is 2,3, and 4) and feature vector $S$ obtained after dimension reduction by the max pooling operation; Finally, module (3) is used to calculate the matching probability for the fused high-level semantics $F$ output by modules (1) and (2) to obtain the final classification result. The following sections explain each part of this method in detail.

### *3.1. Word Vector Based on GloVe*

The traditional word feature representation method uses Word2Vec, but its shortcomings are obvious: the word and vector have a one-to-one relationship, making it difficult to solve synonyms and polysemous words. GloVe is an unsupervised technology that inherits most of the advantages of Word2Vec. It uses global statistics and global prior information and integrates the advantages of the co-occurrence window, which makes it more advantageous in the processing of synonyms and polysemous words and can contain more semantic and grammatical information. The GloVe model does not need to use a neural network for training and can directly use a corpus to calculate word vectors, which is easier to parallelize. In this paper, we used the word "vector GloVe," pre-trained by Stanford University (https://nlp.stanford.edu/projects/glove/ (accessed on 1 June 2022)), and the dimension of the word vector was 100.

### *3.2. Skip-GRU Network Module*

GRU always reads the entire input contents of texts, but for the text classification task, most of the inputs are not necessary for the prediction results. When reading text, the Skip-GRU network module can skip over irrelevant information and produce more accurate predictions. The Skip-GRU model is shown in Figure 2. The Skip-GRU network module consists of three parts: (i) Skip Network; (ii) GRU Network; (iii) Improved Self-Awareness Mechanism. Algorithm 1 shows the construction process of the skip network.

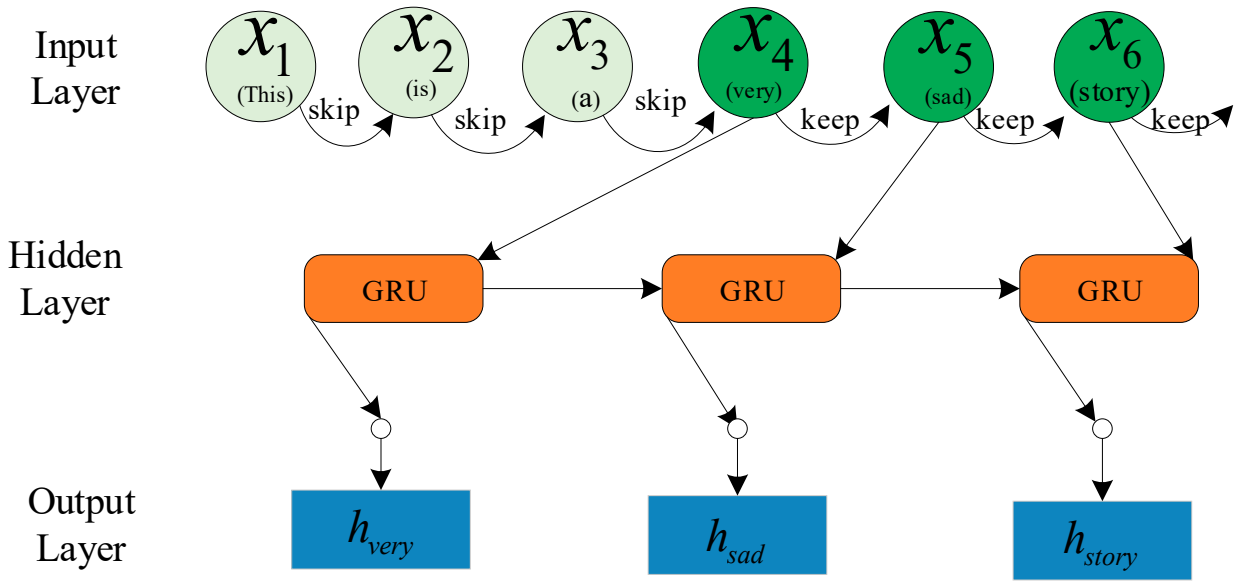


**Figure 2.** Skip-GRU model.

#### 3.2.1. Skip Network

The task of the skip network is to calculate the jump probability before the word vector is input into the GRU network, determine the information to be skipped, and reserve more useful information in GRU according to the calculated jump probability. The skip network architecture is shown in Figure 3.

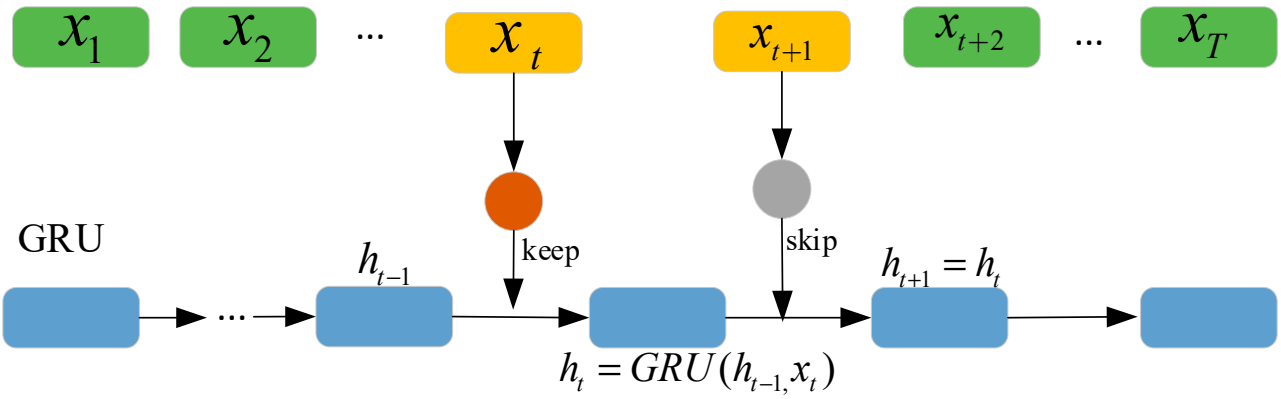


**Figure 3.** Skip-GRU network architecture.

The model is based on the standard GRU. Given an input sequence denoted as $x_1$, $x_2$, ..., $x_T$, or $x_{1:T}$ with length $T$. It can be denoted that $x_t \in R^d$ as the word embedding of the word at position $t$. Before inputting the text information into the GRU network, the text information needs to be input into two layers of a multi-layer perceptron, and the jump probability distribution is calculated by the perceptron. The jump probability is calculated as follows:

$$S_t = RELU(W_1 S + b_1) \tag{1}$$

$$\pi_t = softmax(W_2 S_t + b_2) \tag{2}$$

where $W_1$, $W_2$, $b_1$ and $b_2$ are the weights and biases of the two-layer multilayer perceptron. $S_t$ is the state of the hidden state and $\pi_t$ represents the probability.

#### 3.2.2. GRU Network

GRU, a version of the conventional RNN, is a widely used gated recurrent neural network. The gradient disappearance or explosion issue can be reduced with GRU, much like LSTM, which can also successfully capture the semantic link between lengthy sequences. GRU has two gates: a reset gate and an update gate. The reset gate controls how new input data is integrated with old memory, while the update gate determines how much of the past memory is saved to the current time step.

The standard GRU reads each word sequentially and uses the update function to refresh the hidden state. The unit structure of the GRU is shown in Figure 4. The status update for the GRU network is as follows:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{3}$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{4}$$

$$\widetilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{5}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t \tag{6}$$

where $W_r$, $W_z$, and $W$ denotes the different weight matrices, $x_t$ is the input information of the current moment, $h_{t-1}$ is the hidden state of the previous moment, $r_t$ is the reset gate, $z_t$ is the update gate, and $h_t$ is the output value of the cell.

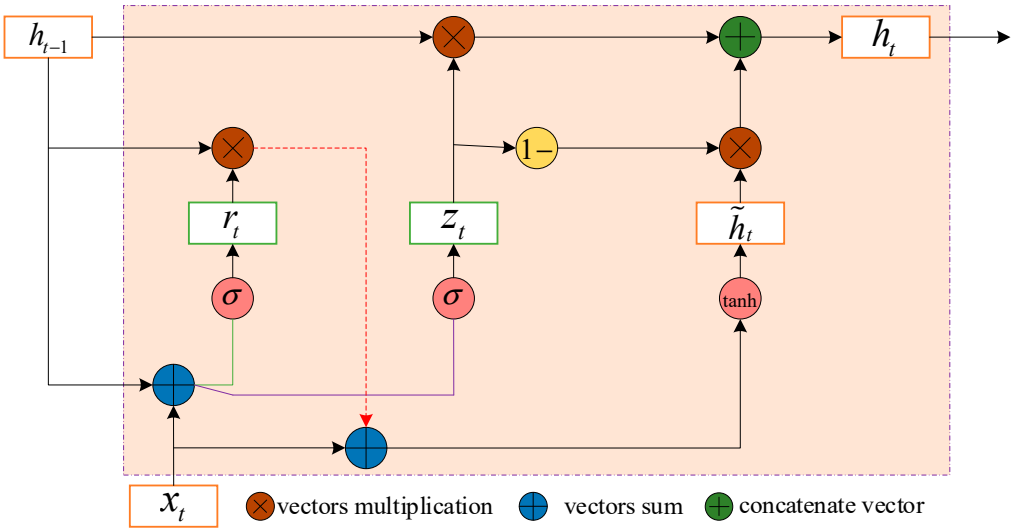


**Figure 4.** The unit structure of GRU.

The skip probability value $\pi_t$ in 3.1.1 determines whether the word can be sent to the GRU network, and 0.5 is chosen as its threshold value. When $\pi_t < 0.5$, the input word is skipped, and the hidden layer is not updated:

$$h_t = h_{t-1} \tag{7}$$

when $\pi_t > 0.5$, it means that the word is more important to the classification result and will be sent to the GRU network. At this time, the hidden state of the GRU network will update as Equations (3)–(6). Algorithm 1 shows the construction of the skip network.

**Algorithm 1** Construction of The Skip Network

**Input:** The text sequence $T$
**Output:** The probability of $x$
**1.** Do convolution operations on $x$ in $T$ to obtain word vector $S$
**2. for** vector $S$ in $T$ **do**
**3.** $S_t = RELU(W_1 S + b_1)$
**4.** $\pi_t = softmax(W_2 S_t + b_2)$
**5.** **if** $\pi_t < 0.5$ then
**6.** skip this word
**7.** **else**
**8.** put $x$ into GRU network
**9.** **end if**
**10. end for**

3.2.3. Improved Self-Attention Mechanism

In the text classification task, each word has a different degree of influence on the classification result. In order to distinguish the importance of each word, a self-attention mechanism layer is introduced to weight the output vector processed by Skip-GRU. The self-attention mechanism is a special variant of the attention mechanism. In order to better understand the principle of the self-attention mechanism, the calculation process of the attention mechanism is firstly analyzed. The attention mechanism can be understood as a mapping function composed of multiple $Query$ and $Key - Value$, and the calculation is shown in the following equations:

$$f(Q, K) = QK^T \tag{8}$$

$$a_i = softmax(f(Q, K)) \tag{9}$$

$$attention(Q, K, V) = \sum a_i V \tag{10}$$

where $Q$ stands for $Query$, $K - V$ stands for $Key - Value$, is the vector-key value, $V$ is the corresponding key value, and $a_i$ is the weight value obtained after normalization by the $softmax$ function.

The addition of the self-attention not only highlights the key features of the text but also yields a more accurate representation of the text feature. In the text classification task, the text vectors at different positions in the input sequence are considered to contribute differently to the output results. For example, the observation window for text vectors trained in the front position is narrow, and the information gained during training is limited, hence the self-attention weights generated from training are frequently larger. In order to avoid this phenomenon, the position weight parameter $Weight$ is introduced to improve the self-attention mechanism. Redistribute the calculated self-attention weight probability value, suitably reduce the weight of the text vector at the front of the training position, and appropriately increase the weight of the text vector at the back of the training position, so as to further optimize the representation of text feature vectors and enhance the expression of text feature ability. The initial value of $Weight$, which is a parameter iterator and a subclass of Tensor, is 1. It is continuously optimized throughout the training phase in order to increase the weight of text characteristics in the back position while decreasing the overall weight of self-attention in the front position. The improved self-attention mechanism assigns weight information as follows:

$$self - attention = softmax(\frac{X \cdot X^T}{\sqrt{d_k}})X \tag{11}$$

$$self - attention_{Weight} = softmax(\frac{X \cdot X^T}{\sqrt{d_k}})X \cdot Weight \tag{12}$$

$$Weight = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mm} \end{pmatrix} \tag{13}$$

where $m$ is the text word length, $X \in R^n$ is the $n$-dimensional vector output from the GRU layer, and $\sqrt{d_k}$ denotes the adjustment factor of the function, which usually indicates the dimensionality of the input vector. The adjustment factor can adjust the inner product of $X \cdot X^T$ to avoid the uneven distribution of the results due to the large gap between the values obtained by the function.

### 3.3. Optimized CNN

CNN is the main method for extracting data features in deep learning, which has the advantage of multi-channel parallelism and setting multiple convolutional kernels to extract features. In this paper, three CNN channels are used for local feature extraction from text, and the parameters of the three channels are independent of each other. In order to enhance the learning ability of CNN and obtain richer text features, a batch normalization layer is introduced. The optimized CNNs consist of convolutional and pooling layers. Algorithm 2 shows the construction of an optimized CNN. The main working process of CNN is shown in Figure 5.

#### 3.3.1. Convolutional Layer

Following the convolution layer's acquisition of feature data from the network's preceding layer, the convolution kernel performs convolution operations from different abstract levels, producing features with local key information. For a convolutional kernel, usually only one class of feature can be extracted, but the content of each type of text is not the same, therefore using different sizes of convolutional kernels can extract richer local

features. In this paper, the kernel size of convolution is set to 2, 3, and 4, and the number of filters is set to 256.

**Algorithm 2** Construction of The Optimized CNN

**Input:** Text vectors $V$
**Output:** Multi-granularity text features
1. Text features $S$ = None
2. $k$ = convolution kernel size (2,3,4)
3. **for** $i$ in $k$ **do**
4. $S_i$ = CNN($V$), $k = i$
5. $S = S + S_i$
6. **end for**

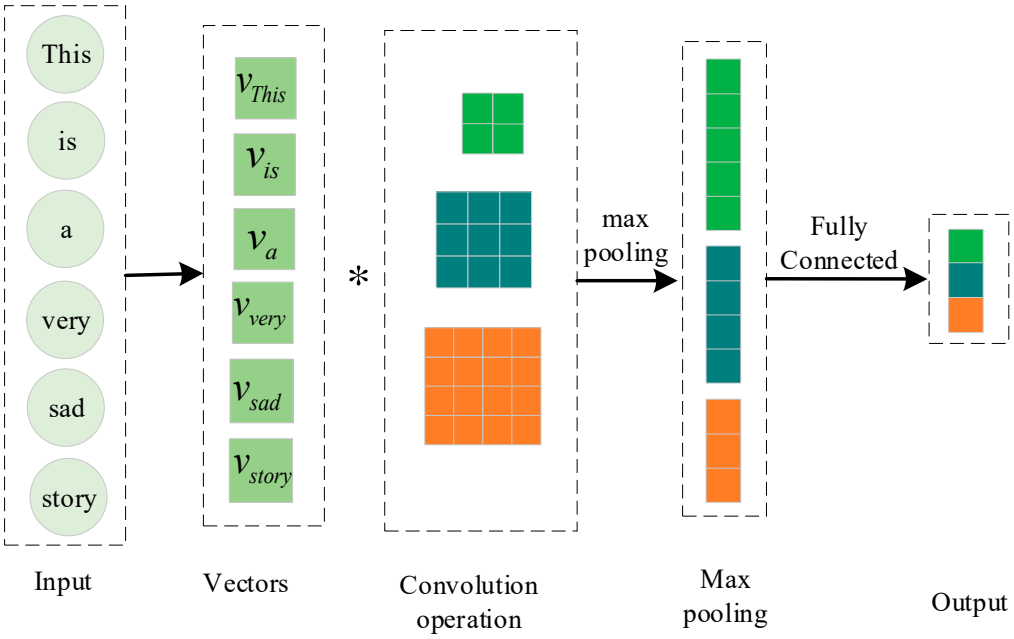


**Figure 5.** The working process of multi-size CNN.

The operation of convolution can be expressed as:

$$S_i = f(W \cdot P_{i:i+h-1} + b) \tag{14}$$

where $S_i$ denotes the $i$th eigenvalue of the text output by the convolution operation, $f()$ is the activation function Relu, $\cdot$ is the dot product of two matrices, $P_{i,j}$ denotes the word vector matrix that is sufficient for the $i$th word to be the $j$th word, and $b$ is the bias.

### 3.3.2. Pooling Layer

There are three common pooling operations: maximum pooling, minimum pooling, and average pooling operations. The feature vector output from the convolution layer usually has a high dimensionality, which can put a large computational burden on the model. The pooling layer can do dimensionality reduction on the information obtained from the convolutional layer, which not only can alleviate the computational pressure problem but also can prevent the model from overfitting. In this paper, the maximum pooling operation is chosen at the pooling layer, which is effective in improving the computational efficiency while maintaining the most significant features as much as possible. After the pooling operation, a fixed-length vector is output:

$$S = \max(S_i) \tag{15}$$

where $S$ is the fixed vector obtained by the pooling operation of the vector from the convolutional layer.

### 3.4. Connection Output Layer

Finally, the softmax classifier is used to calculate the feature vector output by the previous layer and to determine the category to which the text belongs. The probability that the classifier classifies the text into class $j$ is:

$$P(y^i = j|x^i;\theta) = \frac{\exp(\theta_j^T x^i)}{\sum_{n=1}^{k} \exp(\theta_j^T x^i)} \tag{16}$$

where $x$ is the text input, $\theta$ is all the parameters in the training process, and $k$ is the total number of text categories.

## 4. Experiment and Results

This section discusses the data set of the experiment, the establishment of the experimental environment, and the experimental results. First, the data sets are introduced in Section 4.1; then, in Section 4.2, experimental parameter settings and evaluation indexes of experimental results are introduced. The last Section 4.3 not only shows the comparison of experimental results between the proposed model and other advanced models but also conducts ablation experiments as well as experiments on the influence of different text lengths, CNN convolution kernel sizes, and Skip-GRU hidden layer states on text classification results.

### 4.1. Dataset

In order to verify the effectiveness of the text classification model proposed in this paper, relevant experiments were conducted on the following 3 datasets, as shown by the statistics in Table 1.

**Table 1.** Statistics of datasets.

| Dataset | Task | Training/Test | #Classes | Average Length |
|---|---|---|---|---|
| AGNews | News | 120,000/7600 | 4 | 45.00 |
| IMDB | Sentiment | 25,000/25,000 | 2 | 223.27 |
| R8 | News | 5485/2189 | 8 | 65.72 |

AGNews dataset. The AGNews dataset (http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html (accessed on 1 June 2022)) is a collection of news articles collected by academic news search engine ComeToMyHead from more than 2000 news sources in 4 categories. The four categories include the world, science and technology, sports, and business. The AGNews dataset is a balanced short-text dataset, in which the number of each text category is equal, and the number of each text category is equal in the training set and the test set. There are 120,000 training samples and 7600 test samples in the dataset.

IMDB dataset. The IMDB dataset (https://ai.stanford.edu/~amaas/data/sentiment/ (accessed on 1 June 2022)) is developed for the task of binary sentiment categorization of movie reviews with long text and paragraph form. There are an equal number of positive and negative reviews on IMDB. The two categories in the IMDB dataset are denoted "positive" and "negative." The IMDB dataset is a balanced long-text dataset with an equal number of text categories in both the training and test sets. IMDB consists of an equal number of positive and negative reviews, which were split equally between the training set and the test set, with 25,000 comments each.

R8 dataset. The R8 dataset (https://github.com/kk19990709/text-classifier-by-pytorch/tree/main/data (accessed on 1 June 2022)) is a subset of the Reuters news dataset and contains 8 topics. The 8 categories in the R8 data set are: ship, money-fx, grain, acq, trade,

earn, crude, and interest. The R8 dataset is an unbalanced data set in which the number of each text category is equal. There are a total of 5485 training data points and 2189 testing data points.

### 4.2. Experimental Setup

In this paper, the Adam optimizer is used to optimize the model parameters, which uses the same learning rate for different parameters and independently adaptively adjusts the parameters during the training process to speed up the convergence of the model. The learning rate is adjusted by the warmup strategy. On the one hand, this strategy helps to reduce the early overfitting of the model to the mini-batch in the initial stage, and on the other hand, it helps to maintain the stability of the deep model. The parameter information is shown in Table 2.

**Table 2.** Experimental parameters setting.

| Parameters | Value |
|---|---|
| Window size of CNN kernel | (2, 3, 4) |
| Hidden size of GRU | 128 |
| Dropout rate | 0.3 |
| Weight_decay | 0.1 |
| Learning rate | 0.0001 |
| Optimizer | Adam |

In order to verify the performance of different classification methods, this paper evaluates all classification methods based on a confusion matrix. A confusion matrix is a way of capturing and extracting the significance of predictions and true values. False positives (FP) are values that were predicted to be positive but are negative. False negatives (FN) are values that were predicted to be negative but are positive. True positives (TP) are values that are predicted to be positive and are positive. True negatives (TN) are both predicted and observed to be negative [24]. Several indicators can be derived from these metrics, such as Acc (accuracy), P (precision), R (recall), and F1 (f1-score), which are defined as:

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{17}$$

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{18}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{19}$$

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{20}$$

In summary, the variables in the metric are defined as follows:

True positive (TP): Comments that were initially categorized as positive and were projected to be positive by the classifier.

False positive (FP): Comments that were initially categorized as positive but were projected by the classifier to be negative.

True negative (TN): Comments that were initially categorized as negative and were also predicted to be negative by the classifier.

False negative (FN): Comments that were categorized as negative but were predicted as positive by the classifier.

### 4.3. Results Analysis

#### 4.3.1. Performance

Taking accuracy as the evaluation index, the text classification model proposed in this paper is experimentally compared with the following various baseline models:

CNN-based models, including TextCNN [25] and CNN-LSTM [26].

RNN-based models, including LSTM-CNN [27], BiLSTM [28], and LSTM [29].

Semi-supervised models, including SALNET [30]. SALNET includes attention-based LSTM (SALNET1), attention-based LSTM+TextCNN (SALNET2), and attention-based LSTM+BERT (SALNET3).

Adversarial training-based models, include KATG [31] and Lex-AT [32]. KATG includes KATG + TextCNN(KATG1), KATG + LSTM (KATG2), and KATG + BERT (KATG3). Lex-AT includes Lex + TextCNN (Lex-AT1), Lex + LSTM (Lex-AT2), and Lex + BERT (Lex-AT3).

The results are shown in Table 3.

**Table 3.** The main experimental findings (Accuracy/%).

| Method | AGNews | IMDB | R8 |
|---|---|---|---|
| TextCNN | 88.08% | 75.39% | 90.11% |
| LSTM | 87.07% | 84.82% | 90.68% |
| BiLSTM | 87.36% | 85.78% | 91.27% |
| CNN-LSTM | 86.79% | 87.23% | 90.98% |
| LSTM-CNN | 87.89% | 88.73% | 92.32% |
| SALNET1 | 88.22% | 79.00% | - |
| SALNET2 | 89.23% | 80.33% | - |
| SALNET3 | 90.35% | 84.87% | - |
| KATG1 | 92.90% | 89.50% | - |
| KATG2 | 92.80% | 87.90% | - |
| KATG3 | 95.10% | 92.80% | - |
| Lex-AT1 | 92.00% | 88.70% | - |
| Lex-AT2 | 91.90% | 87.40% | - |
| Lex-AT3 | 94.20% | 92.60% | - |
| SA-SGRU (Proposed) | 93.18% | 90.22% | 94.51% |

In-depth analysis of the experimental comparison results in Table 3 reveals that the single network models, such as TextCNN, LSTM, and BiLSTM, do not perform very well because the single network model can only extract the single text feature.

SALNET is a semi-supervised bootstrap learning framework, and it is based on semi-supervised and attention-oriented LSTM, which is insensitive to local key pattern information and performs less well than the SA-SGRU model in terms of classification accuracy. It can be seen that BERT exhibits the best performance compared to SALNet using baseline classifiers such as attention-based LSTM and TextCNN. BERT obtains text representations using a bi-directional transformer encoder and has richer text semantic understanding than attention-based LSTM and attention-based LSTM+TextCNN. Thus, attention-based LSTM+BERT performs better in text classification tasks.

The accuracy of the models based on KATG and Lex-AT adversarial training is no less than 91.50% on AGNews and no less than 87.40% on the IMDB dataset, outperforming semi-supervised self-learning frameworks in classification results. The accuracy of the three models based on KATG adversarial training is higher than that of the three models based on Lex-AT adversarial training. KATG not only uses the previous sentence to guide the generation of adversarial sentences but also proposes a keyword bias-based sampling method to select the sentence containing the biased word as the previous sentence. Compared with the model constructed by Lex-AT, the model constructed by KATG is more capable of capturing the keyword information in the sentences, resulting in greater classification accuracy.

Compared with the above baseline models, except KATG3 and Lex-AT3, the proposed SA-SGRU outperforms the other baseline models on all three datasets.

#### 4.3.2. Ablation Analysis

To validate the effectiveness of each part of the SA-SGRU model, we also conducted ablation experiments for each module on AGNews and IMDB datasets. There are not only comparisons between different modules and the SA-SGRU model, but also comparisons between the GRU+self-attention and CNN modules before the improvement and the corresponding Skip-GRU+improved self-attention and the optimized CNN modules after the improvement. The experimental results are shown in Table 4.

**Table 4.** Effects of each part of SA-SGRU on classification accuracy.

| Method | AGNews | IMDB | R8 |
|---|---|---|---|
| CNN | 86.32% | 78.45% | 90.02% |
| GRU+self-attention | 88.26% | 85.78% | 93.41% |
| Multi-CNN | 90.84% | 80.13% | 93.70% |
| Skip-GRU+improved self-attention | 88.65% | 85.41% | 93.66% |
| SA-SGRU(Proposed) | 93.18% | 90.22% | 94.51% |

Table 4 shows that the improved channels all have some degree of advantage in classification accuracy relative to the unimproved channels, indicating that the performance of each improved part has been improved. Multi-CNN improves by 4.52%, 1.68%, and 3.68% on three datasets in classification accuracy compared with CNN, indicating that the optimized CNN can extract multi-dimensional and richer text features by multi-dimensional convolutional kernels, which well compensates the weakness of CNN with inadequate content extraction by single convolutional kernels. Comparing the classification performances of both the Multi-CNN model and the Skip-GRU+improved self-attention model with the total model SA-SGRU, there is a certain degree of decrease in accuracy, indicating that the modules are complementary to each other. It also shows that both the improved Skip-GRU+improved self-attention module and the optimized CNN module are effective for the classification performance of SA-SGRU.

#### 4.3.3. The Effect of Sequence Length on Model

When dealing with text sequences in text classification tasks, their length is usually fixed. However, since each sentence is of a different length and contains different information, it is important to choose an appropriate text sequence length. In this part, we will explore the impact of different text sequences on the classification of the model. The average length of data in AGNew is 45, the average length of data in IMDB is 223.27, and the average length of data in R8 is 65.72. The sentences in each dataset are set up as five groups of sequences of different lengths. If the text length is less than the predetermined processing length, the zero mark will be padded, and if the text exceeds the predetermined processing length, it will be truncated. Figure 6 shows the three datasets with different text sequence lengths and the corresponding experimental results.

The results are shown in Figure 6. It can be seen from Figure 6a,b that, based on the average length of the sequence in the two short text datasets AGNews and R8, appropriately increasing the length of the text sequence can improve the classification performance of Skip-GRU. Too long text sequences will not only not improve the classification performance, but too long sentence sequences will also cause words with less semantic connection to participate in the derivation of keywords, affect the weight of keyword assignment, and may also lead to degradation of classification performance and an increase in running time. From Figure 6c, we can see that when the sequence length does not reach the average length, the model classification effect increases with the sequence length. When the length of the text sequence exceeds the average length, the classification effect of the model is improved to a certain extent, but the improvement effect is not obvious.

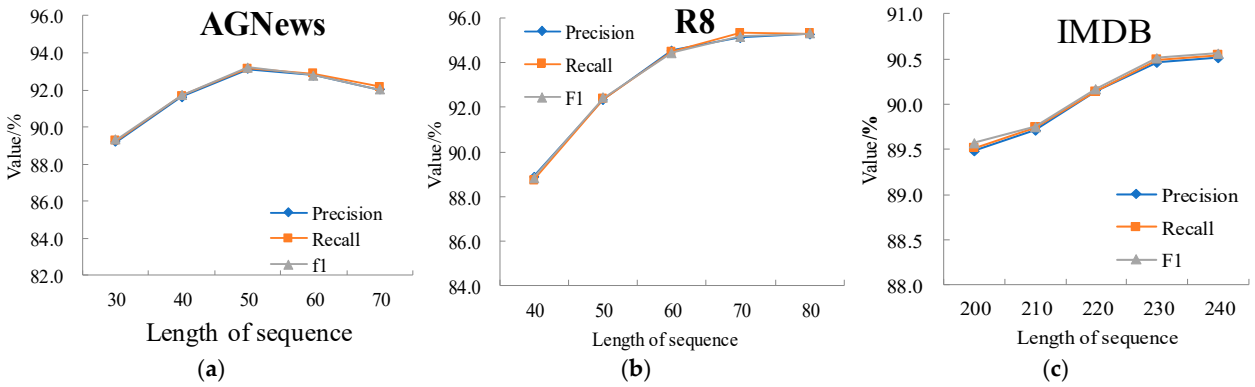


**Figure 6.** Classification performance of different sequence lengths in three datasets. (**a**) represents the influence of different text sequence lengths on model classification in AGNews data set; (**b**) represents the influence of different text sequence lengths on model classification in R8 data set; (**c**) represents the influence of different text sequence lengths on model classification in IMDB data set.

#### 4.3.4. The Effect of CNN Convolutional Kernel Size on Model

In order to verify the influence of convolution kernel size on the final text classification effect, the control variable method was adopted in the setting of model parameters. The hidden layer dimension of Skid-GRU was set to a fixed value of 128 dimensions, and then 5 groups of convolution kernels with different sizes were selected for experimentation. The five groups of convolution kernel data of different sizes selected in the experiment are shown in Table 5, and the experimental results are shown in Figure 7.

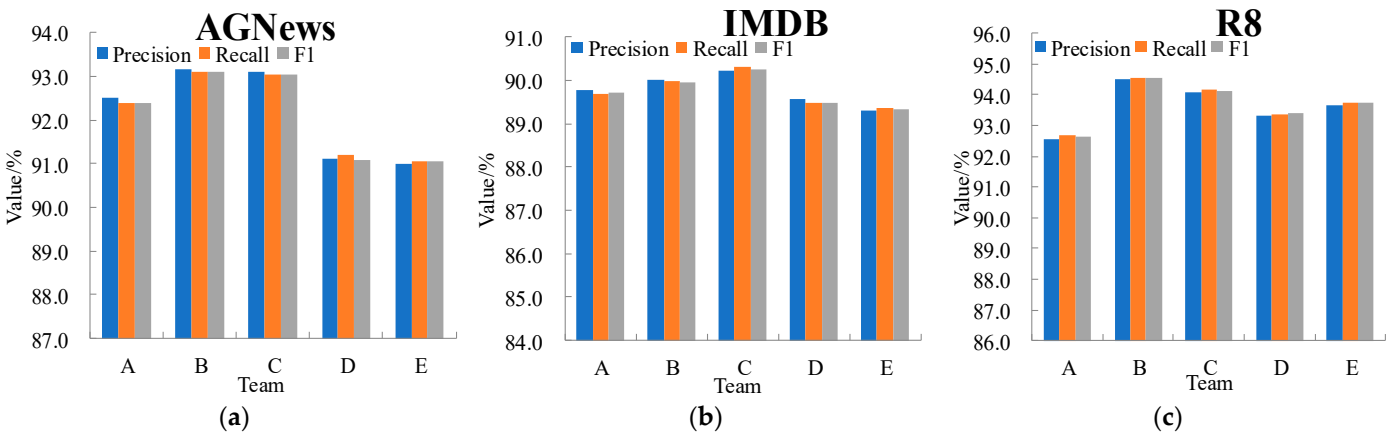


**Figure 7.** Experimental results for different size groups of convolution kernels. (**a**) represents the influence of different convolution kernel sizes on the classification effect of AGNews dataset; (**b**) represents the influence of different convolution kernel sizes on the classification effect of IMDB data sets; (**c**) represents the influence of different convolution kernel sizes on the classification effect of R8 dataset.

**Table 5.** Groups of different sizes of convolutional kernels.

| Group | Convolution Kernel Size |
|---|---|
| A | (1, 2, 3) |
| B | (2, 3, 4) |
| C | (3, 4, 5) |
| D | (2, 3, 4, 5) |
| E | (3, 4, 5, 6) |

As shown in Figure 7, from Figure 7a, it can be seen that when the convolution kernel size is group A and group B, the model performs better on the AGNews dataset than the other groups; from Figure 7b, it can be seen that when the convolution kernel size is the 5 groups of data in Table 5, although the performance of the model on IMDB data is not much different, it performs best in group C; and from Figure 7c, it can be seen that when the convolution kernel size is group A, group C, group D, and group E, the performance of the model on the R8 dataset is not much different, but the model performance is the best in group C. This also proves that there is no fixed set of model parameters applicable to all datasets.

#### 4.3.5. The Effect of Skip-GRU Hidden State Dimension on Model

In this section, we will study the impact of different hidden states of skip GRU on classification. Similar to the size of the CNN convolution kernel, we explore the changes in the classification performance of the model when the hidden layer state dimensions of the skip GRU are set to different sizes. The size of the CNN convolution kernel is fixed at (2, 3, and 4), and the hidden layer state dimension of the skip GRU is adjusted from 32 dimensions; four values of 32, 64, 128, and 256 are selected for experiments. The experimental results are shown in Figure 8. In Figure 8, (a) shows the variation of the accuracy of the model with the hidden layer value on the three datasets, and (b) shows the average time required for the model to train 1 epoch on the training set under different dimensions.

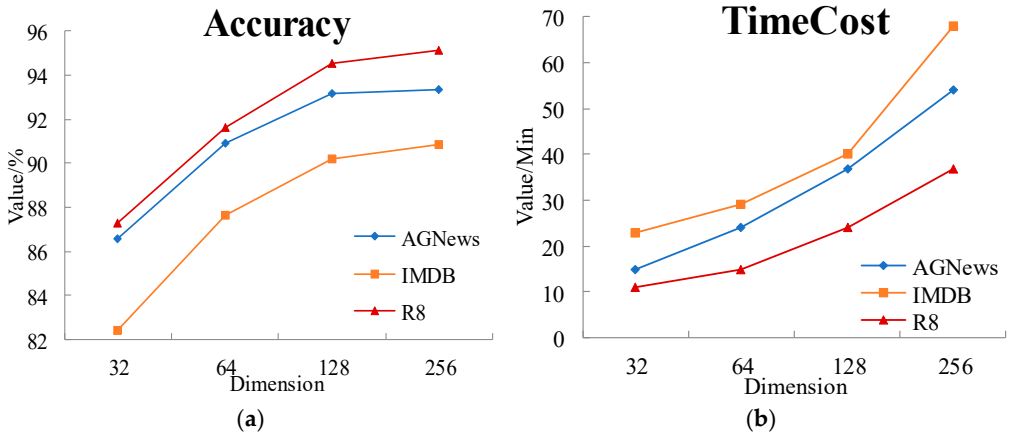


**Figure 8.** Experimental results for different sizes of the skip-GRU hidden state. (**a**) represents the variation of the accuracy of the SA-SGRU model on the three data sets with the value of the hidden layer; (**b**) represents the average time required by the SA-SGRU model to train epoch 1 on the training set under different dimensions.

As can be seen from Figure 8a in Figure 8, with the increase of the hidden layer value, the text classification effect of the model on the three datasets is significantly improved. However, when the dimension reaches 128 dimensions and continues to increase the dimension value, the classification accuracy of the model on the AGNews, IMDB, and R8 datasets is only improved by 0.19, 0.64, and 0.61, respectively. At this time, the classification effect is not significantly improved compared to the 128-dimensional model. From Figure 8b, Figure 8b shows that when the dimension is greater than 128, the model training time increases significantly while the performance remains stable.

## 5. Discussion

Overall, the above experimental results show that our proposed SA-SGRU model can maintain a high level of accuracy in text classification while skipping unimportant information in the text. Compared with other advanced models, the SA-SGRU model

reduces model computation and improves model operation efficiency when skipping text information.

Moreover, the SA-SGRU model is simple to implement. In the SA-SGRU model, the standard GRU and self-attention mechanisms are improved, CNN is optimized, and the advantages of the three are combined to make up for each other's shortcomings: (1) For Skip-GRU, GRU is adopted as the semantic analysis module, which can not only overcome the problem of the gradient explosion of RNNs but also capture semantic information at a greater distance. Since GRU network always reads all the input content of text, and for text classification tasks, most of the input is not necessary for the prediction results, a jump mechanism is set before GRU network, which can skip irrelevant information when reading text and improve the semantic reading efficiency of GRU network. (2) At present, the self-attention mechanism focuses on exploring the influence of each word in a sentence on the semantic level of the sentence and assigning attention weight to each word. However, considering the different contributions of text vectors at different positions in the input sequence to the output results, for example, text vectors trained in the front position have relatively limited information in the training process due to the small observation window. Therefore, the overall weight of self-attention obtained by training will be greater. In order to avoid this phenomenon, the positional parameter *Weight* is introduced to change the self-attention mechanism. In order to further optimize the representation of text feature vectors and enhance the expression ability of text features, the calculated probability value of self-attention weight should be redistributed to appropriately reduce the weight of the text vector at the front of the training position and appropriately increase the weight of the text vector at the back of the training position. (3) In order to make up for the insensitivity of local key information in the Skip-GRU module, CNN with a multi-size convolution kernel is introduced. Usually, only one type of feature can be extracted from a convolution kernel, but the contents of various texts are different. Therefore, CNN with convolution kernels of different sizes can extract richer local features and improve the learning ability of the model.

Both the comparison and ablation experiments with baseline models show that the SA-SGRU model proposed in this paper has certain advantages in performance. However, when BERT is combined with the baseline model, BERT adopts a bidirectional transformer encoder to obtain text feature representation. Compared with GRUs, it has a more powerful text semantic understanding ability. In addition, the effects of text sequence length, CNN convolution kernel size, and Skip-GRU hidden layer dimension on the classification performance of the SA-SGRU model are also discussed.

In our future work, we will mainly focus on the following aspects: (1) designing a parallel computing method to reduce the time spent by the model in extracting features from the word context when the classification accuracy is comparable; (2) combining BERT and deep learning to fully play the role of BERT in text classification; and (3) applying our model to other language datasets besides English.

## 6. Conclusions

For text classification, when reading text, especially long text, a large number of words are irrelevant and can be skipped. Therefore, we started with improving GRU and self-attention mechanisms and proposed a text classification model combining skip-GRU and improved self-attention mechanisms. The model not only includes an improved GRU network and self-attention mechanism but also optimizes CNN to further extract richer text features. Different from the traditional model feature extraction method, we use a jump network mechanism to directly skip the information that is not important to the classification results from the text and then transfer the retained important information into the GRU network, which can effectively capture the semantic association between long text sequences. Considering that the GRU network is not sensitive to key pattern information, we use improved self-attention to reassign weight information to the information captured by the GRU network to enhance the expression ability of text features. Simultaneously, in

order to further enrich the text's features, optimized CNN is used to extract the text's local key information.

We tested the performance of the SA-SGRU model on three public data sets and obtained good experimental results. We also compared the SA-SGRU model with some of the most advanced text classification models based on deep learning. In addition to the KATG3 and Lex-AT3 models, the SA-SGRU model proposed in this paper is superior to other comparison models on the three data sets. After an in-depth study, it was found that BERT, which has stronger textual semantic understanding ability, is used in both the KATG3 and Lex-AT3 models in the text vectorization part, while the SA-SGRU model proposed by us uses GloVe in the text vectorization part. Compared with the BERT model, the SA-SGRU model's text semantic understanding ability in the text vectorization part is not so good, resulting in the final performance of the model being slightly worse than the KATG3 and Lex-AT3 models. In order to enhance the semantic understanding ability of the SA-SGRU model in the text vectorization part, we plan to combine the jump network proposed in this paper with BERT. However, considering the large amount of computation in the BERT model, improving the BERT model and reducing the amount of computation will be the focus of our research.

**Author Contributions:** Funding acquisition, Y.H.; Conceptualization, X.D.; Investigation, J.Y. and Z.H.; Data curation, X.D.; Formal analysis, J.Y. and Z.H.; Methodology, Y.H. and X.D.; Writing—original draft, X.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the Natural Science Foundation of China under Grant No. 61772449, in part by the Natural Science Foundation of Hebei Province (Youth) under Grant No. D2021402043, and in part by the Handan Science and Technology Bureau Foundation under Grant No. 21422093285.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Zhang, Q.; Gao, T.; Liu, X. Public Environment Emotion Prediction Model Using LSTM Network. *Sustainability* **2020**, *12*, 1665. [CrossRef]
2. Xiong, H.X.; Yang, M.T.; Li, Y.Y. A Survey of Information Organization and Retrieval Based on Deep Learning. *Inf. Sci.* **2020**, *38*, 3–10.
3. Liu, Y.; Lapata, M. Learning structured text representations. *Trans. Assoc. Comput. Linguist.* **2018**, *6*, 63–75. [CrossRef]
4. Wang, W.Y.C.; Wang, Y. Analytics in the era of big data: The digital transformations and value creation in industrial marketing. *Ind. Mark. Manag.* **2020**, *86*, 12–15. [CrossRef]
5. Iram, A.; Saeed, H.A.; Wasif, A. Big Data Testing Techniques: Taxonomy, Challenges and Future Trends. *arXiv* **2022**, arXiv:2111.02853v4.
6. Ahad, M.A.; Tripathi, G.; Agarwal, P. Learning analytics for IoE based educational model using deep learning techniques: Architecture, challenges and applications. *Smart Learn. Environ.* **2018**, *5*, 7. [CrossRef]
7. Zhou, L.; Pan, S.; Wang, J.; Vasilakos, A.V. Machine learning on big data: Opportunities and challenges. *Neurocomputing* **2017**, *237*, 350–361. [CrossRef]
8. Manipur, I.; Manzo, M.; Granata, I.; Giordano, M.; Maddalena, L.; Guarracino, M.R. Netpro2vec: A Graph Embedding Framework for Biomedical Applications. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2021**, *19*, 729–740. [CrossRef]
9. Fan, W.; Fan, L. News Text Classification Based on Hybrid Model of Bidirectional Encoder Representation from Transformers and Convolutional Neural Network. In Proceedings of the International Conference on Information Technology and Intelligent Control (CITIC 2021), Guilin, China, 23–25 July 2021.
10. Zeng, F.F.; Li, Y.K.; Xiao, K. Sentence-level fine-grained news classification based on convolutional neural network. *Comput. Eng. Des.* **2020**, *41*, 978–982.
11. Lu, W.; Li, P.C.; Zhang, G.B.; Cheng, Q.K. Recognition of Lexical Functions in Academic Texts: Automatic Classification of Keywords Based on BERT Vectorization. *J. China Soc. Sci. Tech. Inf.* **2020**, *39*, 1320–1329.
12. Devlin, J.; Chang, M.W.; Lee, K.; Kristina, T. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805v2.

13. Gaurav, M. Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better. *arXiv* **2021**, arXiv:2106.08962.
14. Zhong, J.W.; Liu, W.; Wang, S.L.; Yang, H. Review of Methods and Applications of Text Sentiment Analysis. *Data Anal. Knowl. Discov.* **2021**, *5*, 1–13.
15. Deng, J.F.; Cheng, L.L.; Wang, Z.W. Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification. *Comput. Speech Lang.* **2021**, *68*, 101182. [CrossRef]
16. Chen, K.J.; Liu, H. Chinese Text Classification Method Based on Improved BiGRU-CNN. *Comput. Eng.* **2021**, *10*, 1–9.
17. Chen, Q.; Han, L.; Wang, G.S.; Guo, X. Multi-label classification of options based on Seq2seq model of Hybrid Attention. *Comput. Eng. Appl.* **2021**, *57*, 1–10.
18. Hu, J.M.; Fu, W.L.; Qian, W.; Tian, P.L. Research on Policy Text Classification Model Based on Topic Model and Attention Mechanism. *Inf. Stud. Theor. Appl.* **2021**, *44*, 159–165.
19. Feng, Y.; Cheng, Y. Short Text Sentiment Analysis Based on Multi-Channel CNN With Multi-Head Attention Mechanism. *IEEE Access* **2021**, *9*, 19854–19863. [CrossRef]
20. Li, W.J.; Qi, F.; Tang, M.; Yu, Z.T. Bidirectional LSTM with Self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing* **2020**, *387*, 63–77. [CrossRef]
21. Alagha, I. Leveraging Knowledge-Based Features with Multilevel Attention Mechanisms for Short Arabic Text Classification. *IEE Access* **2022**, *10*, 51908–51921. [CrossRef]
22. Du, J.; Huang, Y.; Moilanen, K. Knowledge-aware leap-lstm: Integrating prior knowledge into leap-lstm towards faster long text classification. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 14, pp. 12768–12775.
23. Fiok, K.; Karwowski, W.; Edgar, G.F.; Davahli, M.R.; Wilamowski, M.; Ahram, T.; Awad, A.J.; Zurada, J. Text Guide: Improving the Quality of Long Text Classification by a Text Selection Method Based on Feature Importance. *IEEE Access* **2021**, *9*, 105439–105450. [CrossRef]
24. Jing, W.P.; Song, X.; Di, D.; Song, H. GeoGA T: Graph model based on attention mechanism for geographic text classification. *arXiv* **2021**, arXiv:2101.11424v1.
25. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arXiv* **2014**, arXiv:1408.5882.
26. Zhang, Y.; Yuan, H.; Wang, J.; Zhang, X.J. YNU-HPCC at EmoInt-2017: Using a CNN-LSTM Model for Sentiment Intensity Prediction. In Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Copenhagen, Denmark, 8 September 2017; pp. 200–204.
27. LI, Y.; Wang, X.T.; Xu, P.G. Chinese text classification model based on deep learning. *Future Internet* **2018**, *10*, 113. [CrossRef]
28. Sharfuddin, A.A.; Tihami, M.N.; Islam, M.S. A Deep Recurrent Neural Network with BiLSTM Model for Sentiment Classification. In Proceedings of the International Conference on Bangla Speech and Language Processing (ICBSLP), Sylhet, Bangladesh, 1–4 September 2018.
29. Basnet, A.; Timalsina, A.K. Improving Nepa-Li News Recommendation Using Classification Based on LSTM Recurrent Neural Networks. In Proceedings of the 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), Kathmandu, Nepal, 25–27 October 2018; pp. 138–142.
30. Lee, J.H.; Ko, S.K.; Han, Y.S. SALNet: Semi-supervised Few-Shot Text Classification with Attention-based Lexicon Construction. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 13189–13197.
31. Shen, L.F.; Li, S.S.; Chen, Y. KATG: Keyword-Bias-Aware Adversarial Text Generation for Text Classification. In Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; Volume 36, pp. 11294–11302.
32. Xu, J.J.; Zhao, L.; Yan, H.Q.; Zeng, Q.; Liang, Y.; Sun, X. LexicalAT: Lexical-based Adversarial Reinforcement Training for Robust Sentiment Classification. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 5518–5527.