Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 5594895, 10 pages
https://doi.org/10.1155/2021/5594895

Research Article

Hierarchical Self-Attention Hybrid Sparse Networks for Document Classification

Weichun Huang,[1] **Ziqiang Tao**,[1] **Xiaohui Huang**,[2] **Liyan Xiong**,[2] **and Jia Yu**[2]

[1]*School of Software, East China Jiaotong University, Nanchang 330013, China*
[2]*Department of Information, East China Jiaotong University, Nanchang 330013, China*

Correspondence should be addressed to Weichun Huang; hwc@ecjtu.edu.cn

Received 24 January 2021; Revised 1 April 2021; Accepted 7 April 2021; Published 21 April 2021

Academic Editor: Yu Liu

Document classification is a fundamental problem in natural language processing. Deep learning has demonstrated great success in this task. However, most existing models do not involve the sentence structure as a text semantic feature in the architecture and pay less attention to the contexting importance of words and sentences. In this paper, we present a new model based on a sparse recurrent neural network and self-attention mechanism for document classification. Subsequently, we analyze three variant models of GRU and LSTM for evaluating the sparse model in different datasets. Extensive experiments demonstrate that our model obtains competitive performance and outperforms previous models.

1. Introduction

Text classification is one of the most important subtasks in natural language processing, which can be divided into short text classification and document classification according to the text length. Tradition methods of machine learning are often used for document classification in the past. However, it cannot fully express the semantic information of the text. With the development of deep learning, there are many updated methods to learn word vector representations, which can capture the semantic relations in the vector space. CNN, RNN, and attention mechanism are proven to have strong capabilities in sequence information. Thus, these methods are widely applied to document classification.

More recently, Liu [1] extracted both the forward and backward n-gram features of the text via bidirectional convolutional operations. Pappagari [2] extended fine-tuning procedure of Bert to address one of its major limitations-applicability to inputs longer than a few hundred words for document classification. Yi [3] proposed a local and global context attention (LGCA) model and a multi-context attention (MCA) model to extract text feature. However, most of the updated methods lack the consideration of the importance for different sentences and words. Specifically, it was indicated that part of critical sentences and words in the document have a clear relation to the classification result. Moreover, these methods did not address sentences and words in a document separately, nor did they effectively select informative sentences and words.

Apparently, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been introduced in many hybrid models with elevated results. Although these models perform an outstanding result in several tasks, they introduce an increase in parameters and runtime due to the raising of gate units. Thus, it attempts to prune the external gates parameters to compress RNN models. There are many researchers who designed structure sparse strategies in recurrent neural network model [4–6], but the effort in analyzing the large scale of datasets, especially in text classification, is lacking. In order to reduce computational expense while ensuring accuracy, we try to integrate the sparse strategy into the hybrid RNN model for document classification.

In this paper, we present a new model for the document classification approach to improve the model ability by selecting better representation through self-attention

inspired by [7, 8]. At the same time, we adopt three different sparse strategies to restructuring GRU and LSTM to ensure the effect while reducing redundant parameters. We proposed our model as Hierarchical Self-Attention Hybrid Sparse Network (HSAHSN) based on sparse bidirectional RNNs and self-attention. Our contributions can be concluded as follows:

(1) We adopt self-attention layers to capture a deeper relationship for the contexting importance of words and sentences by hierarchical representation.

(2) We propose three sparse RNN models by hierarchical representation on different scales of datasets, which decreased the parameters and runtime.

(3) We evaluate our models on two document classification datasets, which demonstrates that our model obtains competitive performance and outperforms previous models.

2. Related Work

2.1. Deep Neural Networks for Document Classification. CNN can get similar n-gram information, as the number of convolutional neural network layers increases, the field of view of the convolution will also expand, and wider semantic information can be obtained. Kim [9] adopted multiple filters with different window sizes to extract multiscale convolutional features for text classification. Johnson and Zhang [10] proposed a low-complexity word-level deep convolutional neural network (CNN) architecture for text categorization that can efficiently represent long-range associations in text. Kim and Yang [11] proposed the sequence-to-convolution neural networks (Seq2CNN) and Gradual Weight Shift (GWS) method to stabilize training.

RNN has a superior ability for sequence information, which suits for excavating semantic information and data with sequence characteristics. Wang and Tian [12] incorporated the residual networks [13] into RNN, which makes the model handle a longer sequence. Xu [14] proposed a novel LSTM with a cache mechanism to capture long-range sentiment information. Mikhail and Nikunj [15] proposed the à la carte embedding based on byte-level recurrent language models [16] achieve impressive efficiency on results.

Attention mechanism has been qualified to find out the important information of the sentence without interacting with the distance of words in different positions. Giannis and Antoine [17] represented documents as word cooccurrence networks and proposed an application of the message passing framework for document understanding. Manzil and Guru [18] proposed BigBird with a sparse attention mechanism that reduces this quadratic dependency to linear and showed that BigBird is a universal approximator of sequence functions.

Although recent deep neural networks have achieved great success in document classification, most existing models lack consideration of select task-friendly features on sentences and words by separating document to learning sentence representation. In addition, the traditional attention mechanism overly depends on external information, and we adopt self-attention [19] to capture the internal relation in words and sentences, which can replace sequence-aligned recurrence entirely.

2.2. Sparse Recurrent Neural Networks. A number of researchers have reported the influence in sparse strategies of RNNs unit. The first category method [4, 20]: the strategy of pruning filters is used for network compression. In particular, Xiong and Ling [5] used pruning strategies to preserve important connections during the training phase. Wen [6] decreased the memory requirements of LSTMs by altering the structure of LSTMs. Dey and Salem [21] evaluated three variants of the GRU in recurrent neural networks by reducing parameters in the update and reset gates. The most recent sparse methods on RNN have been applied to many tasks with superior results. We can integrate the sparse strategy into the hybrid RNN model for decreasing parameters and runtime while ensuring accuracy in document classification.

2.2.1. Hierarchical Self-Attention Hybrid Sparse Network (HSAHSN). The architecture of our model has shown two main components: sparse word encoder and sparse sentence encoder, consisting of sparse bidirectional recurrent neural network and self-attention. The following two subsections describe the overall framework of our model in Section 2.1 and how we apply sparse methods in RNN cells in Section 2.2.

2.3. Framework of HSAHSN. The HSAHSN model has three parts: Word embedding, sparse word encoder, and sparse sentence encoder. Figure 1 shows the main structure of the model, which concludes CNN, sparse bidirectional RNN, and self-attention mechanism.

In the word embedding layer, we adopt Fasttext [22] as the word embedding initialization. To extract different n-gram word representations in a sentence, we adopt different sizes of filter in the CNN layer after word embedding to extract more features and advance generalization ability. Given a sentence with words $o_{it}, t \in [1, n]$ and an embedding matrix W_e trained by Fasttext, words are embedded to vectors through embedding matrix W_e. Subsequently, embedding words are convoluted by various sizes of filter W_f and get a concatenate sentence feature matrix c_i^*:

$$\begin{aligned} x_{it} &= W_e o_{it}, \quad t \in [1, n], \\ c_i^* &= f\left(W_f, x_i\right), \end{aligned} \tag{1}$$

where $f(\cdot)$ is a composite function including two cascaded operations: a convolution and a rectified linear unit (ReLU).

In sparse word encoder, we consider three distinct variants of sparse bidirectional RNN cell units to capture sequence features through forward and back directions. Therefore, it can incorporate contextual information in the annotation.A sparse bidirectional RNN contains the forward sparse RNN $\overrightarrow{f}$ which reads the sentence feature c_i^* from w_{i1}

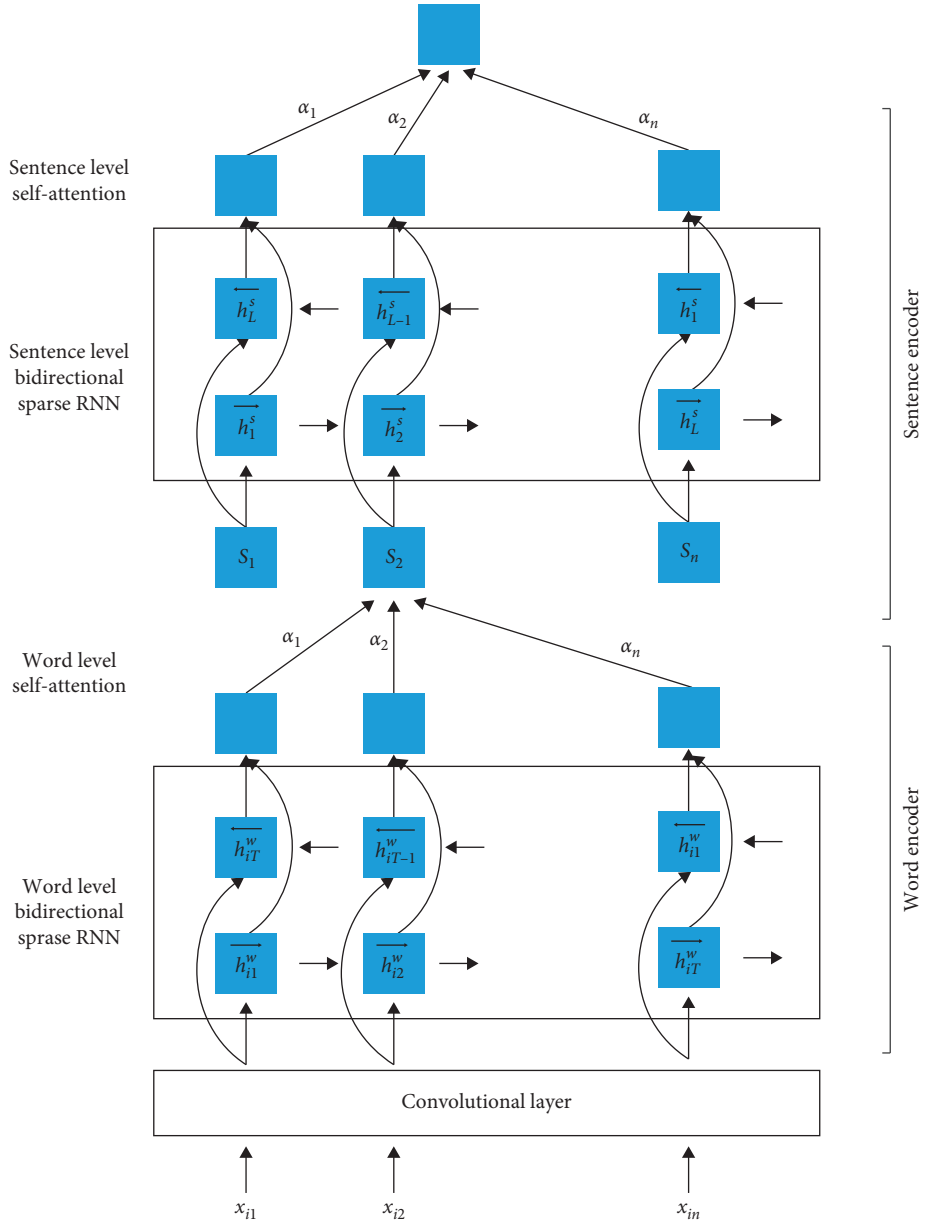

Figure 1: Framework of our HSAHSN model.

to w_{iT} and backward sparse RNN $\overleftarrow{f}$ which reads from w_{iT} to w_{i1}:

$$\begin{aligned} \overrightarrow{h_{it}^{w}} &= \overrightarrow{\text{SRNNs}}(w_{it}), \quad t \in [1, T], \\ \overleftarrow{h_{it}^{w}} &= \overleftarrow{\text{SRNNs}}(w_{it}), \quad t \in [T, 1], \end{aligned} \tag{2}$$

and concatenating the forward hidden state $\overrightarrow{h_{it}^{w}}$ and backward hidden state $\overleftarrow{h_{it}^{w}}$ can explicate the word w_{it} in a bidirectional direction, that is, $h_{it}^{w} = [\overrightarrow{h_{it}^{w}}, \overleftarrow{h_{it}^{w}}]$, which summarizes the information for the whole sentence.The importance of words in the sentence is different. Hence, the attention mechanism has been introduced to solve this problem. However, the traditional attention mechanism overly depends on external information and the effect is depending on the initialization and training of parameters.

To address the issue of extracting the internal relation of words and sentences in a document, we use the self-attention mechanism to evaluate the importance of words in a sentence. h_{it}^{w} are packed together into matrices Q_{it}, K_{it}, and V_{it}. Specifically,

$$s_i = \text{soft max}\left(\frac{Q_{it} K_{it}^{T}}{\sqrt{d_k}}\right) V_{it}, \tag{3}$$

where v is the sentence vector that capture all the information of words in a sentence. In this way, we can calculate the importance of words in a sentence and get the relevance of the words in the sentence directly.

Given a sentence vector s_i, we can get a document vector in the same way. Firstly, the s_i input into bidirectional sparse RNNs:

$$\begin{aligned} \overrightarrow{h_i^s} &= \overrightarrow{\text{SRNNs}}(s_i), \quad t \in [1, L], \\ \overleftarrow{h_i^s} &= \overleftarrow{\text{SRNNs}}(s_i), \quad t \in [L, 1]. \end{aligned} \tag{4}$$

Subsequently, the model can obtain the bidirectional semantic information of each sentence in the document by concatenating $\overrightarrow{h_i^s}$ and $\overleftarrow{h_i^s}$, that is, $h_i^s = [\overrightarrow{h_i^s}, \overleftarrow{h_i^s}]$. h_i^s is a vector that comprises nearby sentences but still concentrates on sentence i. However, we need to pay attention to the sentence that implies classifying the document exactly. We once again adopted the self-attention mechanism to obtain significant information at sentence level vector h_i. Then, h_i are packed together into matrices Q_i, K_i, and V_i:

$$s = \text{soft max}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i, \tag{5}$$

where s is the document vector that captures all information of the sentence vector in a document. Finally, we feed s into a fully connected layer with softmax and we can get a probability distribution over classes.

2.4. Sparse RNN Module. In order to decrease training time and parameter amount while ensuring effectiveness, we adopt three sparse strategies for the RNN model. GRU and LSTM capture the state of sequences through gating structure, which can alleviate the gradient disappearance or explosion in traditional RNN for long sequence samples. Gating units were established to compute the sequential information flow and each gate parameter will be updated through overall network information. Thus, the contemporary state condition can absorb the information of preceding status and current input. However, the gating signals flow, which is the key of the situation of the network, probably involves redundancy information and it is possible to affect the comprehension of the model. In this study, we adopt three different variants of gating strategy for GRU and LSTM:

Variant 1. Each gate unit only calculates the previous hidden state h_{t-1} with weight U and bias b. We called variant 1 GRU1 and LSTM1.

GRU1:

$$z_t = \sigma(U_z h_{t-1} + b_z), \tag{6a}$$

$$r_t = \sigma(U_r h_{t-1} + b_r). \tag{6b}$$

LSTM1:

$$f_t = \sigma\left(U_f h_{t-1} + b_i\right), \tag{6c}$$

$$i_t = \sigma(U_i h_{t-1} + b_i), \tag{6d}$$

$$o_t = \sigma(U_o h_{t-1} + b_o). \tag{6e}$$

Variant 2. Each gate unit only calculates the previous hidden state h_{t-1} with out bias b. We called variant 2 GRU2 and LSTM2.

GRU2:

$$z_t = \sigma(U_z h_{t-1}), \tag{7a}$$

$$r_t = \sigma(U_r h_{t-1}). \tag{7b}$$

LSTM2:

$$f_t = \sigma\left(U_f h_{t-1}\right), \tag{7c}$$

$$i_t = \sigma(U_i h_{t-1}), \tag{7d}$$

$$o_t = \sigma(U_o h_{t-1}). \tag{7e}$$

Variant 3. Each gate unit only calculates bias b without the previous hidden state h_{t-1}. We called variant 3 GRU3 and LSTM3.

GRU3:

$$z_t = \sigma(b_z), \tag{8a}$$

$$r_t = \sigma(b_r). \tag{8b}$$

LSTM3:

$$f_t = \sigma\left(b_f\right), \tag{8c}$$

$$i_t = \sigma(b_i), \tag{8d}$$

$$o_t = \sigma(b_o). \tag{8e}$$

In GRU or LSTM architecture, the recurrent hidden state can be expressed as follows:

GRU:

$$h_t = g(W x_t + U h_{t-1} + b). \tag{9a}$$

LSTM:

$$\tilde{c}_t = g(W_c x_t + U_c h_{t-1} + b_c), \tag{9b}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \tag{9c}$$

$$h_t = o_t \odot g(c_t), \tag{9d}$$

where x_t is a k-dimensional vector at t step. h_t is a n-dimensional vector and n can be treated as the output size. W and U are the parameters for calculating x_t and h_t, respectively, and it will add bias b in the end. Subsequently, W, U, and b can be deduced to be an $n \times k$, $n \times n$, and $n \times 1$ matrix, respectively. The total parameters of recurrent hidden state can be calculated as $n^2 + n \times k + n$.

In GRU cell unit, there are two gates named update gate z_t and reset gate r_t. They have the same parameter structure as the previous recurrent gating unit. Specifically, in this case, the total parameters in GRU are equivalent to $3 \times (n^2 + n \times k + n)$ with recurrent gate when the input and output dimensions are k and n, respectively. In the LSTM cell unit, there are three gates named forget gate f_t, input gate i_t, and output gate r_t. Total parameters in LSTM can be calculated as $4 \times (n^2 + n \times k + n)$.

Thus, the above three strategy parameters can be calculated in Table 1.

3. Results and Discussion

In this section, we give the properties of the datasets and experimental settings in Section 3.1 and Section 3.2, respectively. Subsequently, we show our evaluation results on two datasets in Section 3.3. During the course of training, we plot and analyze the effect of variant models on convergence in Section 3.4. Moreover, the number of parameters for sparse bidirectional RNN and runtime in variant models are recorded in Section 3.5 and Section 3.6.

3.1. Datasets. We evaluated our model on two document classification datasets: IMDb and Yelp 2018. The IMDb dataset is composed of 25 K movie reviews for training data and test data, respectively, wherein the classification includes positive/negative reviews. Yelp 2018 includes 5 M full review text data about users' ratings from 0 to 5 stars for the comments of stores and services. To further explore the effects of variants below, we implement them based on different scale datasets. We split samples for 90% as train data and 10% as test data.

3.2. Implementation Details

3.2.1. Input. Word embedding adopted Fasttext to training with 200 as embedding dimension. The input document text is separated into sentences with padding fixed length 15 and the sentence is padded to a fixed length of 50.

3.2.2. Architecture Configuration. The model is implemented with Keras. We adopt 3 different window sizes of filters for the convolution layer. Clear configuration of sparse word encoder and sparse sentence encoder is included in Tables 2 and 3. The classification layer is a fully connected MLP with a ReLU activation function and softmax output.

3.2.3. Training Settings. We use Adam [23] optimizer with 64 as a batch size. The learning rate is initially set to 0.001. The training process lasts at 30 and 40 epochs on IMDb and Yelp 2018 datasets, respectively.

3.3. Result Comparison and Analysis. In this section, we evaluate the HSAHSN model on two document classification datasets, which are IMDB and Yelp 2018, for three different variants of GRU and LSTM. Baseline models are distinguished with different RNN units, called HSAHN. In the subsequent description, we refer to three variants of GRU as GRU1, GRU2, and GRU3 and three variants of LSTM as LSTM1, LSTM2, and LSTM3, respectively. The results are listed in Table 4.

Table 4 shows the accuracy over three variants of GRU and LSTM with the same configuration setting. There are trends in our data to suggest that GRUs exhibit better accuracy performance over LSTMs about 1.89% to 4.44% in IMDb and 0.22% to 0.5% in Yelp. Simultaneously, after a series of experiments, it is noted that the ability of regular models is elevated after pruning in the IMDb dataset, which achieves 95.69% in HSAHSN + GRU3. Although baseline model HSAHN + GRU presents 73.48% as the best result in Yelp, HSAHSN + GRU2 also exhibits a near result to 73.28%, which is only decreased by about 0.2%. However, HSAHSN + GRU2 highly reduces the amount of parameter and runtime.

Compared with sparse models in Table 4, HSAHN + GRU shows the best performance on the Yelp dataset. However, sparse models showed a very similar accuracy with HSAHN + GRU in Yelp. Moreover, sparse models improve upon baseline models by 2% to 4% in the IMDb dataset. This shows that our sparse methods can effectively drop redundancy information in the gate unit of RNN, which can elevate the comprehension ability of sequence information while reducing the model complexity.

Table 5 shows the experimental results comparing with several genres of the popular model, wherein models with "$*$" contain attention mechanism. Our results are outperformed with various models below in Yelp 2018 and achieve a state of the art, wherein the proposed model obtained an accuracy of 73.48%. Moreover, our model was elevated slightly after pruning in IMDb, which enhanced about 0.68% compared with HAHNN while reducing the parameters.

Experimental result shows that our model outperforms HAHNN by 0.2% with self-attention in Table 5 and our models also give superior results to other models with attention mechanism, which shows that our proposed method can capture the contexting importance of words and sentences in documents.

3.4. Convergence Analysis. To experimentally verify the convergence of variant models of RNN, we plot the loss and accuracy over time with different epochs when models are trained and tested in Figures 2 and 3, respectively.

Figure 2 summarizes the results of loss and accuracy which show comparable performance among three variates of GRU and LSTM. Comparing with other models, HSAHSN + GRU3 achieves 95.85% accuracy particularly. During the course of training, GRU2 is shown to a similar performance with LSTM2. Specifically, the training loss of LSTM2 and GRU2 is elevated by about 0.1 and 0.4, respectively, compared with other models. Simultaneously, the accuracy is decreased by 0.15 and 0.03. The variant 2 model presents an inferior convergence and error estimate compared with other models apparently.

From Figure 3, all GRU variants appear to exhibit comparable accuracy performance in Yelp. Three variants of GRU and LSTM exhibit lower performance in the interval of 0.2% to 0.26% and 0.28% to 0.53% comparing with the original GRU and LSTM model, respectively.

In Figures 2 and 3, we have discovered that bias exhibits an exclusive role in models. While gate units are disposed bias, it will be suppressed the comprehension ability of

Table 1: Total parameters in variant RNNs models.

Model	GRU			
	Variant 1	Variant 2	Variant 3	Normal
Parameter	$3n^2 + n \times (k + 3)$	$3n^2 + n \times k$	$n^2 + n \times (k + 3)$	$3(n^2 + n \times k + n)$
Model	LSTM			
	Variant 1	Variant 2	Variant 3	Normal
Parameter	$4n^2 + n \times (k + 4)$	$4n^2 + n \times k$	$n^2 + n \times (k + 4)$	$4(n^2 + n \times k + n)$

Table 2: Sparse word encoder setting.

Sparse word encoder parameter	Setting
Dropout rate	0.1
RNN output size	50
Activate function	ReLU
Self-attention output size	100

Table 3: Sparse sentence encoder setting.

Sparse sentence encoder parameter	Setting
Dropout rate	0.1
RNN output size	50
Activate function	ReLU
Self-attention output size	100
Kernel regularizer	L2

Table 4: Results in variants models.

Architecture	Models	Yelp 2018	IMDB
Baseline	HSAHN + GRU	73.48	91.75
	HSAHN + LSTM	73.23	88.21
Sparse GRUs	HSAHSN + GRU1	73.22	93.63
	HSAHSN + GRU2	73.28	93.12
	HSAHSN + GRU3	73.22	95.69
Sparse LSTMs	HSAHSN + LSTM1	73.00	92.18
	HSAHSN + LSTM2	72.75	91.23
	HSAHSN + LSTM3	72.94	91.25

Table 5: Results in classification accuracy.

Models	Yelp 2018	IMDB
HAHNN* [2]	73.28	95.17
byte mLSTM7 [12]	—	92.2
Bigbird* [15]	72.16	95.2
BERT-ITPT-FIT [24]	70.58	—
LSTM-reg [25]	68.7	—
MPAD-path* [14]	—	91.84
HMAN* [26]	73.4	
H-CRAN* [27]	73.0	
CapsNet [28]		89.72
Ours	73.48	95.69

models such as GRU2. Bias can give offset compensation when the input distribution is not zero as the center and the stochastic gradient descent is implicitly used to carry information about the network state. These may explain the relative success in using the bias alone in the gate signals.

3.5. Parameter Scale for Sparse RNN. The aim of this section is to evaluate the number of parameters in different RNN variants and the results are listed in Table 6.

Table 6 shows the comparison of the parameters in variants of GRU and LSTM. We set the output dimension

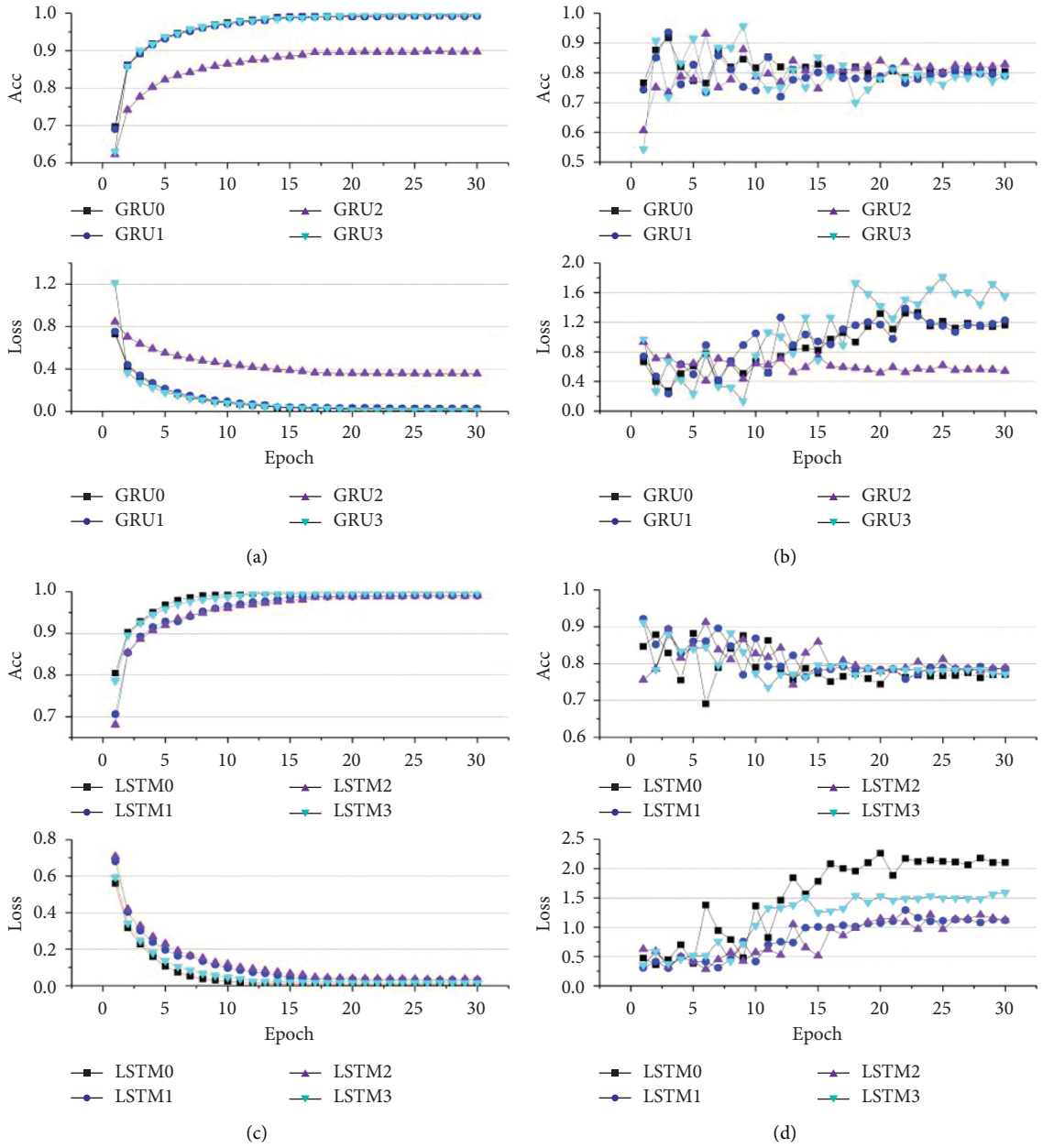

Figure 2: Loss and accuracy of variant RNNs models in IMDb dataset. (a) Training loss and accuracy of HSAHSN + GRU in IMDb. (b) Testing loss and accuracy of HSAHSN + GRU in IMDb. (c) Training loss and accuracy of HSAHSN + LSTM in IMDb. (d) Testing loss and accuracy of HSAHSN + LSTM in IMDb.

as 50 in bidirectional RNNs to calculate the parameters. After comparing three variants sparsity methods, it is found that the third method exhibited the highest sparsity rate of 66.17% and 74.44% for GRU and LSTM, respectively. The reduction of parameters indicates the decrease of FLOPs, thereby reducing the calculation time. We conclude that the parameters do directly interact with the runtime in Table 7.

3.6. Runtime Comparison. In order to assess runtime for various models in an epoch, we record the runtime in different scales of datasets as Table 7.

Table 7 indicates the runtime for an epoch in the variants of GRU and LSTM in IMDb and Yelp. It considerably decreases the relative runtime by 24.03% to 34.42% and 32.94% to 50.59% for GRU and LSTM, respectively, in IMDb. Then, in comparison with IMDb, the result shows the same effect on runtime in Yelp. The runtime is decreased from 29.22% to 38.19% in GRU and from 15.17% to 27.36% in LSTM. Our findings seem to demonstrate that RNN models exhibit a strong position in overall runtime. Specifically, the parallel computing ability is limited due to the special structure of RNN, which is the reason of RNN taking up a lot of computing time in the hybrid model.

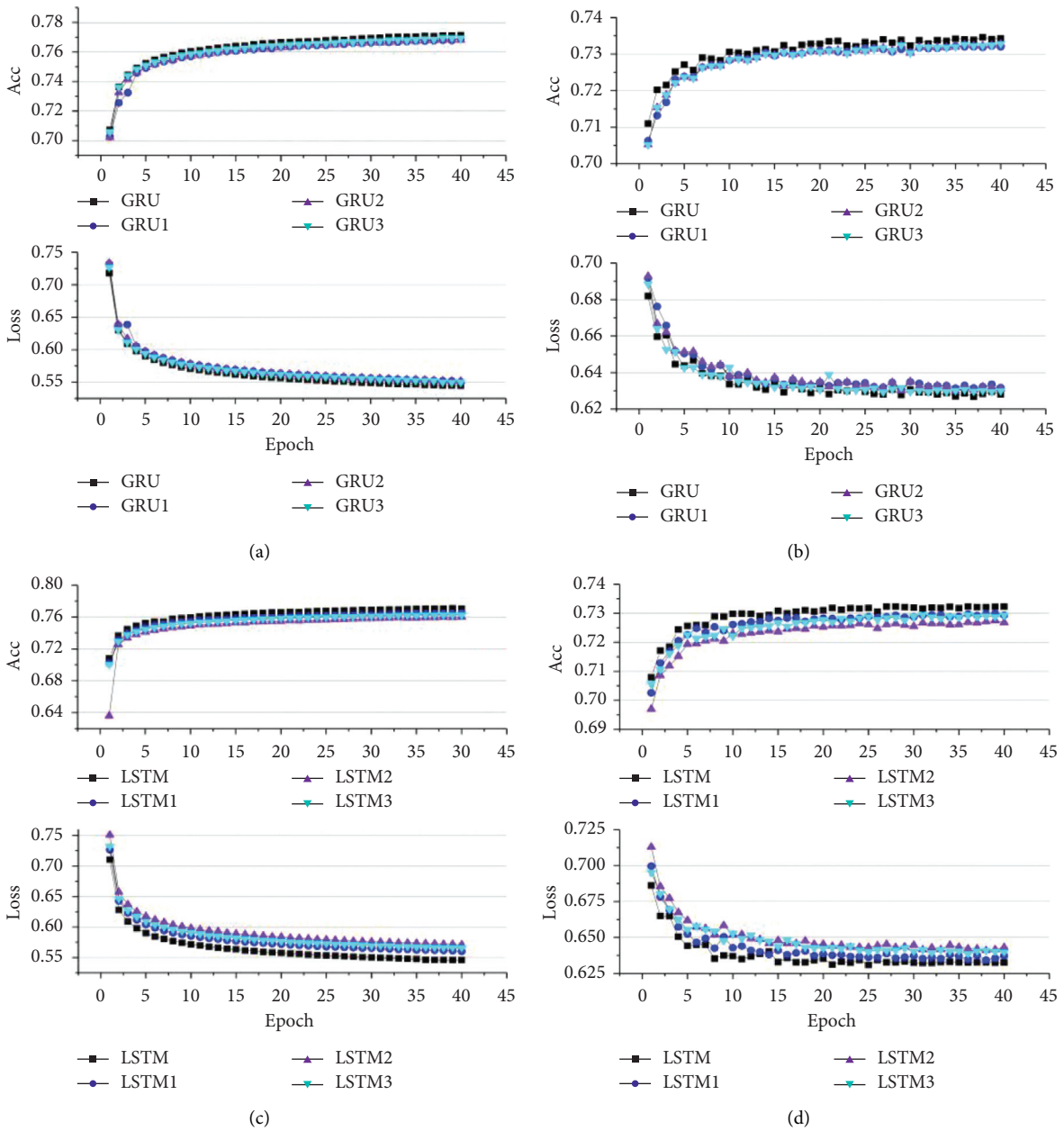

Figure 3: Training loss and accuracy of variant RNNs models in Yelp dataset. (a) Training loss and accuracy of HSAHSN + GRU in Yelp. (b) Testing loss and accuracy of HSAHSN + GRU in Yelp. (c) Training loss and accuracy of HSAHSN + LSTM in Yelp. (d) Testing loss and accuracy of HSAHSN + LSTM in Yelp.

Table 6: Parameters comparison with variants models.

	Param	Pruning rate		Param	Pruning rate
GRU	79800	—	LSTM	106400	—
GRU1	47000	41.26%	LSTM1	57200	46.24%
GRU2	46600	41.60%	LSTM2	56600	46.80%
GRU3	27000	66.17%	LSTM3	27200	74.44%

Table 7: Runtime comparison.

IMDb				Yelp			
Model	Runtime	Model	Runtime	Model	Runtime	Model	Runtime (s)
GRU	154	LSTM	255	GRU	5472	LSTM	6377
GRU1	117	LSTM1	171	GRU1	3873	LSTM1	5413
GRU2	112	LSTM2	151	GRU2	3720	LSTM2	4945
GRU3	101	LSTM3	126	GRU3	3382	LSTM3	4632

4. Conclusions

In this paper, we propose the HSAHSN for document classification. The method is based on sparse RNN and self-attention mechanisms in the word and sentence level. We evaluate our models on Yelp 2018 and IMDb datasets for classification and adopt three sparse variants for GRU and LSTM to assess the effectiveness of models. The proposed model improves the text comprehension ability more than previous models on Yelp 2018 and IMDb. We also analyzed the number of parameters, the runtime, and the loss of two datasets in different sparse models.

Data Availability

The data are available at http://www.kaggle.com/luisfredgs/hahnn-for-document-classification.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grants nos. 61967006, 62062033, and 62067002; the Education Department Project of Jiangxi Province under Grants nos. GJJ190316 and GJJ181547; the Natural Science Foundation of Jiangxi Province under Grant no. 20192ACBL21006; and Science and Technology Project of Jiangxi Provincial Department of Transportation 2021X0011.

References

[1] L. Bin and Y. Guosheng, "Chinese document classification with Bi-directional convolutional language model," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1785–1788, Xi'an, China, September 2020.

[2] R. Pappagari, P. Zelasko, J. Villalba et al., "Hierarchical transformers for long document classification," in *Proceedings of the 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 838–844, Sentosa, Singapore, December 2019.

[3] Y. Liu, H. Yuan, and S. Ji, "Learning local and global multi-context representations for document classification," in *Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1234–1239, Beijing, China, November 2019.

[4] H. Li, A. Kadav, I. Durdanovic et al., "Pruning filters for efficient conv nets," in *Proceedings of the International Conference on Learning Representations*, pp. 1–13, Toulon, France, April 2017.

[5] L. Xiong, X. Ling, X. Huang, H. Tang, W. Yuan, and W. Huang, "A sparse connected long short-term memory with sharing weight for time series prediction," *Institute of Electrical and Electronics Engineers Access*, vol. 8, pp. 66856–66866, 2020.

[6] W. Wen, Y. He, S. Rajbhandari et al., "Learning intrinsic sparse structures within long short-term memory," in *Proceedings of the International Conference on Learning Representations*, pp. 1–14, Vancouver, BC, Canada, April 2018.

[7] Z. Yang, D. Yang, C. Dyer et al., "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, San Diego, CA, USA, June 2016.

[8] J. Abreu, L. Fred, D. Macêdo, and C. Zanchettin, "Hierarchical attentional hybrid neural networks for document classification," in *Proceedings of the Artificial Neural Networks and Machine Learning-ICANN 2019: Workshop and Special Sessions*, pp. 396–402, Munich, Germany, September 2019.

[9] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751, Doha, Qatar, October 2014.

[10] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 562–570, Uppsala, Sweden, July 2017.

[11] T. Kim and J. Yang, "Abstractive text classification using sequence-to-convolution neural networks," Article ID 07745, 1805 pages, 2018, https://arxiv.org/abs/1805.07745.

[12] Y. Wang and F. Tian, "Recurrent residual learning for sequence classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 938–943, Austin, TX, USA, November 2016.

[13] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Honolulu, HI, USA, July 2017.

[14] J. Xu, D. Chen, X. Qiu et al., "Cached long short-term memory neural networks for document-level sentiment classification," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1660–1669, Austin, TX, USA, November 2016.

[15] M. Khodak, N. Saunshi, Y. Liang et al., "A La carte embedding: cheap but effective induction of semantic feature vectors," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 11–22, Melbourne, Australia, July 2018.

[16] A. Radford, R. Jozefowicz, and I. Sutskever, "Learning to generate reviews and discovering sentiment," in *Proceedings of the International Conference on Learning Representations*, Vancouver, BC, Canada, April 2018.

[17] G. Nikolentzos, A. Tixier, and M. Vazirgiannis, "Message passing attention networks for document understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 8544–8551, New York, NY, USA, February 2020.

[18] Z. Manzil, G. Guruganesh, A. Dubey et al., "Big bird: transformers for longer sequences," Article ID 14062, 2020, https://arxiv.org/abs/2007.14062.

[19] V. Ashish, S. Noam, P. Niki et al., "Attention is all you need," in *Proceedings of the Conference and Workshop on Neural Information Processing Systems*, pp. 6000–6010, Long Beach, CA, USA, December 2017.

[20] P. Molchanov, S. Tyree, T. Karras et al., "Pruning convolutional neural networks for resource efficient inference," in *Proceedings of the International Conference on Learning Representations*, pp. 24–26, Vancouver, BC, Canada, April 2017.

[21] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proceedings of the IEEE 60th International Midwest Symposium on Circuits and Systems*, pp. 1597–1600, Boston, MA, USA, August 2017.

[22] A. Joulin, É Grave, P. Bojanowski et al., "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 2, pp. 427–431, Valencia, Spain, April 2017.

[23] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, Banff, AB, Canada, April 2014.

[24] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune Bert for text classification?, lecture notes in computer science," in *Proceedings of the China National Conference on Chinese Computational Linguistics*, pp. 194–206, Kunming, China, October 2019.

[25] A. Adhikari, A. Ram, R. Tang et al., "Rethinking complex neural network architectures for document classification," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 4046–4051, Minneapolis, MN, USA, June 2019.

[26] Y. Huang, J. Chen, S. Zheng et al., "Hierarchical multi-attention networks for document classification," *International Journal of Machine Learning and Cybernetics*, vol. 20219 pages, 2021.

[27] J. Du, L. Gui, Y. He, R. Xu, and X. Wang, "Convolution-based neural attention with applications to sentiment classification," *Institute of Electrical and Electronics Engineers Access*, vol. 7, pp. 27983–27992, 2019.

[28] J. Kim, S. Jang, E. Park, and S. Choi, "Text classification using capsules," *Neurocomputing*, vol. 376, pp. 214–221, 2020.