SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2021, Vol. 64 222102:1–222102:11
https://doi.org/10.1007/s11432-019-2744-2

Dual-axial self-attention network for text classification

Xiaochuan ZHANG[1], Xipeng QIU[2,3*], Jianmin PANG[1], Fudong LIU[1] & Xingwei LI[1]

[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou* 450001*, China;*
[2]*Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai* 201203*, China;*
[3]*School of Computer Science, Fudan University, Shanghai* 201203*, China*

Received 20 September 2019/Revised 19 November 2019/Accepted 26 December 2019/Published online 25 November 2021

Abstract Text classification is an important task in natural language processing and numerous studies aim to improve the accuracy and efficiency of text classification models. In this study, we propose an effective and efficient text classification model which is based on self-attention solely. The recently proposed multi-dimensional self-attention significantly improved the performance of self-attention. However, existing models suffer from two major limitations: (1) the previous multi-dimensional self-attention models are quite time-consuming; (2) the dependencies of elements along the feature axis are not taken into account. To overcome these problems, in this paper, a much more computational efficient multi-dimensional self-attention model is proposed, and two parallel self-attention modules, called dual-axial self-attention, are applied to capture rich dependencies along the feature axis as well as the text axis. A text classification model is then derived. The experimental results on eight representative datasets show that the proposed text classification model can obtain state-of-the-art results and the proposed self-attention outperforms conventional self-attention models.

Keywords text classification, dual-axial self-attention, feature-axial dependency

Citation Zhang X C, Qiu X P, Pang J M, et al. Dual-axial self-attention network for text classification. Sci China Inf Sci, 2021, 64(12): 222102, https://doi.org/10.1007/s11432-019-2744-2

1 Introduction

In this paper, we focus on an important natural language processing (NLP) task, text classification, which is widely used in many fields, such as importance-performance analysis [1], sentiment analysis [2,3] and article recommendation [4]. The research scope of text classification mainly focuses on two aspects: selecting the best features and designing the best machine learning model.

For feature selection, most techniques are based on words, i.e., an input text is represented as a sequence of words. Considering the inherent characteristics of English words and in order to handle out-of-vocabulary words, some studies apply character-level inputs to models and obtain improvement compared with word-level inputs [5,6].

As for text classification models, naive Bayes based linear models are often considered as baselines [7]. Because naive Bayes is efficient and simple, many studies continue to advance the development of naive Bayes [8–10]. Another conventional model to text classification uses a TF-IDF (term frequency-inverse document frequency) vector of the given text as an input feature to a subsequent model. Applying n-gram information to TF-IDF is considered to improve the performance of the text classification model [5].

Recently, models based on neural networks have become increasingly popular. An essential challenge for neural networks based text classification models is to capture contextual dependencies of the whole text. Many solutions for this task utilize convolutional neural network (CNN) [5, 6, 11] and recurrent neural network (RNN) [6, 12, 13] to capture the local and long-term dependencies, respectively.

Self-attention, a special case of attention (for details please see Subsection 2.1), has raised substantial interest in recent years for their high interpretability in dependencies modeling, high parallelism in computation and high flexibility in sequence length handling. Self-attention models the dependency between

* Corresponding author (email: xpqiu@fudan.edu.cn)

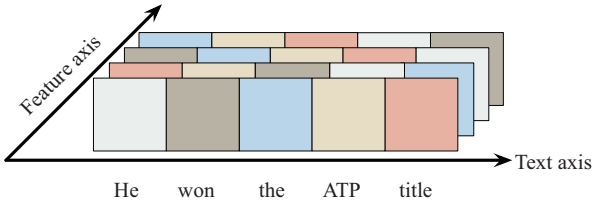

Figure 1 (Color online) Illustration of the text and the feature axes. The colored blocks denote the vectorized representations of the text ("He won the ATP title").

two tokens by applying the softmax function to the outputs of the score function. In the general self-attention, so called single-dimensional self-attention in this paper, the output of the score function is a scalar. Recently, Refs. [14, 15] proposed the multi-dimensional self-attention by extending the output of the score function from a scalar to a vector which further improves the performance of self-attention in various tasks.

Although self-attention has achieved great successes in a broad range of NLP tasks, there are still some aspects remaining to be improved.

(1) The previous multi-dimensional self-attention models are still time consuming in practice;

(2) The dependencies along the feature axis (illustrated in Figure 1) have not been taken into account in previous models. However, we find that elements on the feature axis are not independent of each other in most cases.

In this paper, we propose a text classification model which is based on a novel self-attention model solely. In specific, we first offer a light-weight multi-dimensional self-attention model. Based on this model, the dual-axial self-attention is then proposed, which contains two self-attention modules, i.e., the text-axial self-attention module and the feature-axial self-attention module, used to model the dependencies along the text and the feature axis, respectively. Based on the proposed self-attention, we finally build a text classification model. We conduct experiments on eight representative datasets. Experimental results demonstrate that our proposed self-attention is superior to conventional self-attention models and can obtain higher accuracy than state-of-the-art models in text classification tasks. The contributions of this paper can be summarized as follows.

(1) We propose an effective and efficient text classification model, which outperforms the state-of-the-art models in terms of accuracy.

(2) We propose a novel multi-dimensional self-attention model, which can significantly reduce the computational complexity compared with previous models [14, 15].

(3) We utilize the dual-axial self-attention to explore dependency along the feature axis in addition to the text axis. To the best of our knowledge, this is the first study of the feature-axial dependencies modeling in NLP tasks.

(4) In the text-axial self-attention, we design an attenuation factor to model the attenuation of dependencies with the increase of relative distance. The experimental results show that the use of the attenuation factor improves the performance of the text-axial self-attention.

2 Background

2.1 Attention model

In a broad range of NLP tasks, attention model is widely used to model relationship among tokens from variable-length sequences. Attention model can be formally described as follows. Given three sequences $Q \in \mathbb{R}^{d_1 \times N}$, $K \in \mathbb{R}^{d_1 \times N}$, $V \in \mathbb{R}^{d_2 \times N}$, which are called query, key and value sequences respectively, the output $H = [h_1, h_2, \ldots, h_N] \in \mathbb{R}^{d_2 \times N}$ is derived as follows:

$$h_i = \mathrm{att}\left((K, V), q_i\right) = \sum_{j=1}^{N} \alpha_{ij} v_j = \sum_{j=1}^{N} \mathrm{softmax}\left(s\left(k_j, q_i\right)\right) v_j, \tag{1}$$

where $\alpha_{i,j}$ is often called connection weight from the i-th element to the j-th element and $s(k_j, q_i)$ is the score function. The common score functions are summarized in Table 1 [16–18].

Table 1 Common score functions[a)]

Name	Score function
Additive [16]	$s\left(k_j, q_i\right) = v^{\mathrm{T}} \tanh\left(W q_i + U k_j\right)$
General [17]	$s\left(k_j, q_i\right) = {q_i}^{\mathrm{T}} W k_j$
Dot Product [17]	$s\left(k_j, q_i\right) = {q_i}^{\mathrm{T}} k_j$
Scaled Dot Product [18]	$s\left(k_j, q_i\right) = \frac{{q_i}^{\mathrm{T}} k_j}{\sqrt{d_1}}$

a) W, U, v are learnable parameters and d_1 is the dimension of q_i and k_j.

To model contextual dependencies of the same sequence, self-attention specifies the three sequences Q, K and V generated by three linear transformations on the same input sequence $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{d \times N}$:

$$Q = W_Q X, \tag{2}$$

$$K = W_K X, \tag{3}$$

$$V = W_V X. \tag{4}$$

Gated-Attention [19] is a variant of attention model, in which the output H (in (1)) of general attention serves as a gate matrix and the final output $\tilde{X}$ is generated by

$$\tilde{X} = H \odot X, \tag{5}$$

where $\odot$ denotes element-wised multiplication.

2.2 Motivations

Firstly, inspired by Gated-Attention, element-wised multiplication between the output of single-dimensional self-attention and the original input can be regarded as another approach to multi-dimensional self-attention. We expect that this approach can accelerate computation compared with the previous ones, while maintaining high performance on text classification tasks.

Secondly, in general NLP models, a piece of text is represented as a matrix firstly, which is constructed by a sequence of word embeddings, as illustrated in Figure 1. Then, various neural networks, including attention models, can be used to model dependencies among word embeddings. Actually, the text matrix can also be regarded as consisting of a fixed number of feature vectors and each feature vector is constructed by elements of a specific dimension of word embeddings in the text. In this regard, self attention can be applied to the feature axis hopefully, in addition to the text axis. Thus we try to model element dependencies along both text and feature axes in the proposed self-attention and expect that the introduction of the feature axial dependencies will improve the performance of the self-attention. By coincidence, we find a similar idea which has been exploited in computer vision tasks by applying self-attention to both spatial and channel dimensions [20].

3 Dual-axial self-attention model

In this section, we introduce a novel self-attention model, called dual-axial self-attention, which is used for modeling dependencies along both text and feature axes. An overview of the proposed model is given in Subsection 3.1. Then, a light-weight multi-dimensional self-attention is given in Subsection 3.2, which is the basis of our model. Finally, we introduce the three components of the proposed model respectively, i.e., the text-axial self-attention in Subsection 3.3, the feature-axial self-attention in Subsection 3.4 and the fusion gate in Subsection 3.5.

3.1 Overview

As illustrated in Figure 2, the dual-axial self-attention contains three components: two parallel multi-dimensional self-attention modules and a fusion gate. The two self-attention modules, called text-axial self-attention and feature-axial self-attention, take the same sequence $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{d \times N}$ as input, and exploit rich dependencies along text and feature axes respectively. The final output of the dual-axial self-attention is produced in fusion gate by combining outputs from the two self-attention modules.

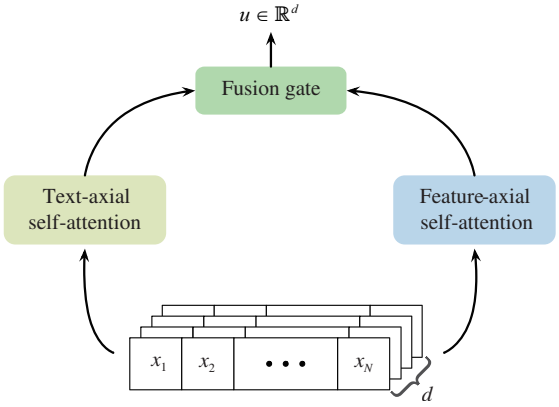

Figure 2 (Color online) The structure of the dual-axial self-attention.

3.2 Multi-dimensional self-attention

The general idea of our multi-dimensional self-attention is to first utilize single-dimensional self-attention to generate an attention-weight matrix and then obtain output by applying element-wise multiplication between the attention-weight matrix and the original input. In the following we shall explain this process step by step.

Given $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{d\times N}$ as the input, we first apply the three linear transformations given in (2)–(4) on X. By constraining $d_1 = d_2 = d$, we obtain three matrices $\{Q, K, V\} \in \mathbb{R}^{d\times N}$.

Then, we select Dot Product [17] as score function, and obtain the attention-weight matrix $W = [w_1, w_2, \ldots, w_N] \in \mathbb{R}^{d\times N}$ by the following operation:

$$w_i = \mathrm{att}\left((K, V), q_i\right) = \sum_{j=1}^{N} \alpha_{ij} v_j = \sum_{j=1}^{N} \mathrm{softplus}\left(s\left(k_j, q_i\right)\right) v_j. \tag{6}$$

It can be seen that we use softplus function in (6) instead of softmax function, which is the only difference compared with (1).

The reasons for not adopting softmax function are mainly based on the following two points. Firstly, owing to the power of exponential, softmax tends to exaggerate distributions, making α_{ij} excessively concentrate on a single element. This property, called unimodality, hinders self-attention from capturing information of the whole input. Secondly, the assumption that $\sum_{j=1}^{N} \alpha_{ij} = 1$ needs further discussion and the strong normalization provided by this condition may also lead to less differentiation among the attention weights (w_i in (6)). Although multi-head self-attention [18] compensates the problem of concentrating on a single element, it cannot solve the second problem.

Softplus is defined as

$$\mathrm{softplus}(x) = \log(1 + \mathrm{e}^x). \tag{7}$$

Softplus is a monotonically increasing function, which maps score function $s\left(k_j, q_i\right) \in \mathbb{R}$ to $\alpha_{ij} \in \mathbb{R}+$, denoting the zoom factor of v_j. Unlike softmax, when $s(k_j, q_i)$ is negative, softplus maps it to a positive number which is closed to 0, and when $s(k_j, q_i)$ is positive, softplus is almost an identity mapping. As a result, the distribution of α_{ij} in (6) is no longer unimodal, making score function consider the whole input rather than concentrate too much on a single element. Further more, the removal of normalization from α_{ij} provided by softmax also leads to significant differentiation of attention weights (w_i in (6)).

Finally, we get the output $H = [h_{ij}] \in \mathbb{R}^{d\times N}$ by

$$h_{ij} = \mathrm{softplus}(w_{ij}) \times x_{ij}. \tag{8}$$

Overall, on the basis of single-dimensional self-attention, only one more step of matrix multiplication (given in (8)) is added to the proposed multi-dimensional self-attention. As a result, the proposed multi-dimensional self-attention is much more computationally efficient than the previous ones.

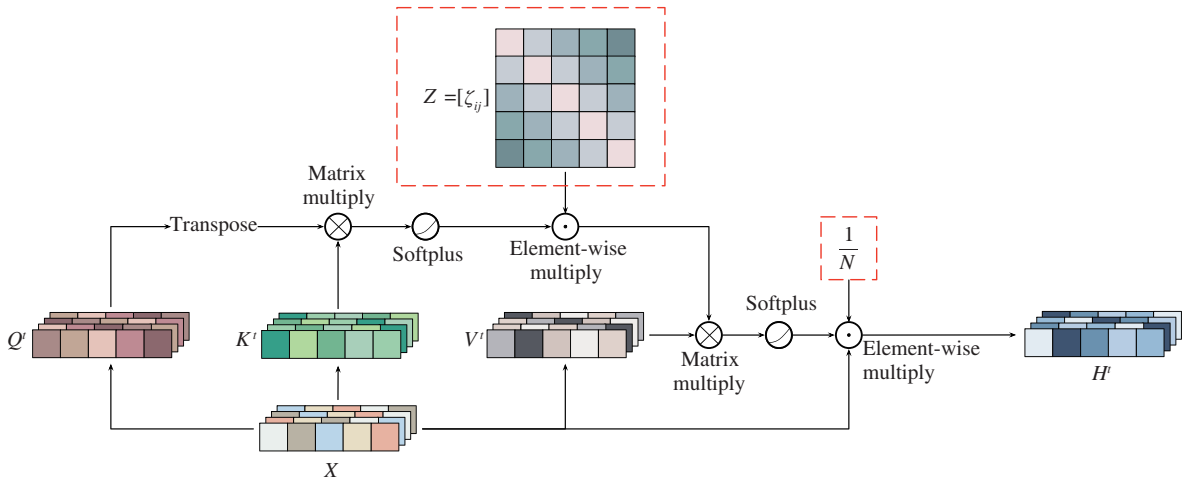

Figure 3 (Color online) Text-axial self-attention module. Note: the two red-dotted boxes are variations on the proposed multi-dimensional self-attention. For the purpose of parallelism, we integrated ζ_{ij} into a matrix $Z = [\zeta_{ij}] \in \mathbb{R}^{N\times N}$.

3.3 Text-axial self-attention

We apply the proposed multi-dimensional self-attention on the text axis to model the context relationship. In consideration of some intrinsic properties in the text-axial dependencies modeling, two variations are applied to the proposed multi-dimensional self-attention. The structure of the text-axial self-attention is given in Figure 3 where the variations are highlighted in two red-dotted boxes.

Specifically, given a sequence $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{d\times N}$, we first feed it into three linear transformations shown in (2)–(4) to generate three matrices Q^t, K^t and V^t, respectively.

The first variation to the proposed multi-dimensional self-attention in Subsection 3.2 is that we introduce an attenuation factor to model the relationship between the dependency and the relative position. From a linguistic intuition, as the relative distance increases, the dependency between two tokens on the text axis should attenuate, but the speed of attenuation should slow down. This implies that the attenuation factor should be a function whose first order derivative is negative and second order derivative is positive. For this reason, we design the attenuation factor ζ_{ij} as follows:

$$\zeta_{ij} = \frac{1}{\log(w|i-j|+c)}, \tag{9}$$

where i, j denote position of two elements on the text axis; w and c are two scalar parameters and both of them are set to e in this study. Hence, Eq. (6) can be modified as follows:

$$w_i^t = \mathrm{att}\left((K^t, V^t), q_i^t\right) = \sum_{j=1}^{N} \alpha_{ij}^t v_j^t = \sum_{j=1}^{N} \mathrm{softplus}\left(s\left(k_j^t, q_i^t\right)\right) \zeta_{ij} v_j^t. \tag{10}$$

Remark 1. Note that similar concerns have been shown in some latest studies. For instance, Ref. [21] offered a solution which considered dependencies among a fixed count of neighbouring tokens instead of the whole input. Ref. [22] formulated a similar hypothesis that precise relative position information was not useful beyond a certain distance. Compared with their studies, the proposed attenuation factor can model the relationship between the dependency and the relative distance through a more soft way.

The second variation to the multi-dimensional self-attention is that each element of the final output is multiplied by a normalization factor μ. For the text-axial self-attention, as illustrated in (10), every element in the attention-weight matrix W^t is a summation over the sequence direction. Thus, normalization is needed to handle various sequence lengths. In specific, we set the normalization factor $\mu = N^{-1}$, and multiply it to every element in the final output. Hence, we get the final output $H^t = [h_{ij}^t] \in \mathbb{R}^{d\times N}$ of the text-axial self-attention by modifying (8) to

$$h_{ij}^t = \mathrm{softplus}(w_{ij}^t) \times x_{ij} \times \mu. \tag{11}$$

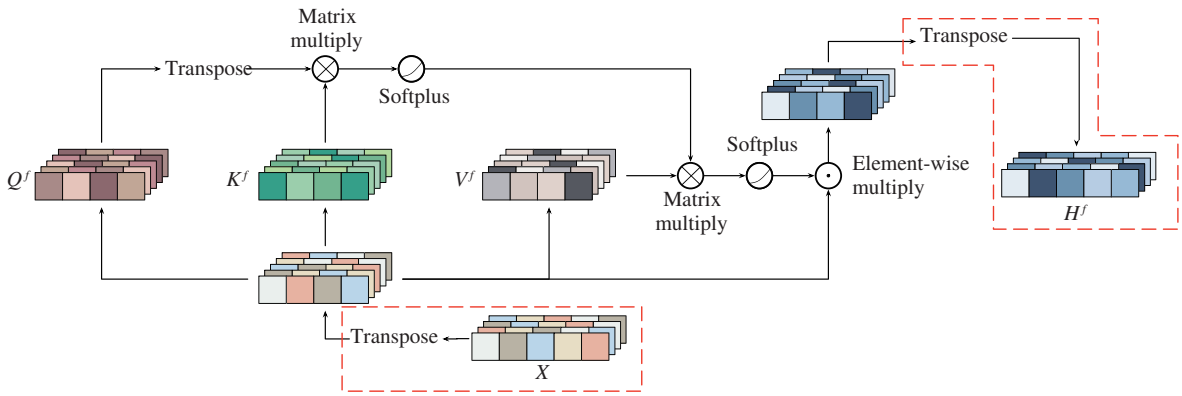

Figure 4 (Color online) Feature-axial self-attention module. Note: the two red-dotted boxes are variations on the proposed multi-dimensional self-attention.

3.4 Feature-axial self-attention

Similar to the text-axial self-attention, the feature-axial self-attention is an application of the proposed multi-dimensional self-attention to the feature axis. Compared with the text-axial self-attention, the relative distance makes no difference to the dependency between two elements for the feature-axial self-attention. Besides, the feature dimension is fixed while the text is not. Hence, the attenuation factor, as well as the normalization factor in the text-axial self-attention, is not taken into consideration in the feature-axial self-attention.

As illustrated in Figure 4, to apply the multi-dimensional self-attention to the feature axis, we only need to apply two transposition operations for the input and the output matrices respectively. In this study, we use H^f to denote the final output of the feature-axial self-attention module.

3.5 Fusion gate

Fusion gate combines outputs of the two self-attention modules and aggregates along the text axis to form the final output $u \in \mathbb{R}^d$ of the dual-axial self-attention. The combination is accomplished by an element-wise weighted summation between H^t and H^f, i.e.,

$$f_i = \text{sigmoid}\left(W^f h_i^f + W^t h_i^t + b\right), \tag{12}$$

$$u = \tanh\left(\sum_{i=0}^{N}\left(f_i \odot h_i^t + (1 - f_i) \odot h_i^f\right)\right), \tag{13}$$

where $\odot$ denotes element-wise multiplication; $\{W^f, W^t\} \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^d$ are the learnable parameters of fusion gate.

4 Text classification model

Because self-attention model is much faster than CNN and RNN, recent studies have focused on using self-attention model solely, and achieved great successes in a broad range of NLP tasks [14, 18, 21–23]. Inspired by their successes, we present the text classification model solely based on the proposed dual-axial self-attention model.

The proposed text classification model consists of the following three components.

(1) Embedding layer: mapping words of the input sentence into vectors of real numbers without any pre-trained word embeddings.

(2) Dual-axial self-attention layer: capturing the key part of the embedded sentence along both text and feature axes.

(3) Fully connected (FC) layer: collecting the outputs from the dual-axial self-attention layer and making a final decision.

Table 2 Descriptive statistics of the datasets used in our experiments

Datasets	Topic	Class	Train sample	Test sample
AG	News classification	4	120000	7600
Sogou	News classification	5	450000	60000
DBP	Wikipedia article classification	14	560000	70000
Yelp.F	Sentiment analysis	5	650000	50000
Yelp.P	Sentiment analysis	2	560000	38000
Amz.F	Sentiment analysis	5	3000000	650000
Amz.P	Sentiment analysis	2	3600000	400000
Yah.A	Questions and answers categorization	10	1400000	60000

In this study, we add bigram information to the input, select the sigmoid function as the activation function in the FC layer, and use softmax normalization before the final output. In our design, the training objective is based on categorical cross-entropy. We minimize the loss function in the training set using the Adam gradient descent algorithm [24], which computes adaptive learning rates for each parameter. The weights are all initialized according to Glorot's scheme [25], while biases are initialized with zeros.

5 Experiments

5.1 Datasets

We employ the same eight datasets from [5], and use the accuracy to evaluate the performance of the proposed text classification model. These datasets contain popular topics that use text classification, including news classification, sentiment analysis, Wikipedia article classification and questions and answers categorization. The scale of samples in these datasets ranges from hundreds of thousands to several million. Table 2 presents a summary.

Remark 2. Although statistical tests are used in some previous studies to compare models' performance, like [26, 27], they are not necessary for this study. This is mainly because the datasets we employed are large enough, which means the distribution of samples is close to the distribution of the real data. In this concern, the previous models evaluated on these datasets did not involve statistical tests either [5, 6, 12, 13, 28]. Besides, we calculate the average result of three tests to avoid fluctuations of results.

5.2 Baselines

We select several representative models to serve as baselines. They are the bag of words (BoW) [5], n-grams [5], n-grams TF-IDF variant [5], character level convolutional model (char-CNN) [5], character based convolution recurrent network (char-CRNN) [6], very deep convolutional network (VDCNN) [11], fastText [28], discriminative long short term memory (Discrim-LSTM) [12] and sliced recurrent neural network (SRNN) [13]. For thorough comparison, to analyze the contribution by each principal component of the dual-axial self-attention better, we implement eight extra baselines to compare with our proposed model.

• **Convention self-attention models.** To show the superiority of our proposed self-attention model over the convention self-attention models, we select four types of conventional single-dimensional self-attention models whose score functions are shown in Table 1 and the previous multi-dimensional self-attention (MDSA) [14, 15].

• **Dual-axial SA w/ softmax.** To prove the rationality of substituting softmax function with the softplus function (in (6)), we compare our proposed model with a modified one equipped with softmax function.

• **Text-axial SA.** To prove the effectiveness of introducing the feature-axial dependencies, we replace the dual-axial self-attention in the model by the text-axial self-attention. That is, we ignore the dependencies along the feature axis.

• **Text-axial SA w/o attenuation factor.** To prove the importance of attenuation factor in the text-axial self-attention, we use the text-axial self-attention instead of the dual-axial self-attention in the text classification model and remove the attenuation factor in the text-axial self-attention, i.e., eliminate the relative distance information of the text-axial self-attention.

Table 3 Test accuracies on eight datasets

	Model	AG	Sogou	DBP	Yelp.F	Yelp.P	Amz.F	Amz.P	Yah.A
Previous models	BoW [5]	88.8	92.9	96.6	58	92.2	54.6	90.4	68.9
	n-grams [5]	92	97.1	98.6	56.3	95.6	54.3	92	68.5
	n-grams TF-IDF [5]	92.4	**97.2**	**98.7**	54.8	95.4	52.4	91.5	68.5
	char-CNN [5]	87.2	95.1	98.3	62	94.7	59.5	94.5	71.2
	char-CRNN [6]	91.4	95.2	98.6	61.8	94.5	59.2	94.1	71.7
	VDCNN† [11]	91.3	96.8	**98.7**	**64.7**	95.7	**63**	**95.7**	73.4
	fastText, bigram [28]	**92.5**	96.8	98.6	63.9	**95.7**	60.2	94.6	72.3
	Discrim-LSTM [12]	92.1	94.9	**98.7**	59.6	92.6	–	–	**73.7**
	SRNN⋆ [13]	92.0	96.0	98.1	62.3	95.4	60.4	95.0	72.3
Self-attention models	Additive [16]	**92.4**	97.3	98.7	61.3	**95.5**	58.7	94.1	72.9
	General [17]	91.8	**97.4**	98.7	61.6	95.3	58.7	94.1	73.0
	Dot Product [17]	91.9	97.3	98.7	61.6	95.4	58.5	94.1	73.1
	Scaled Dot Product [18]	91.7	97.3	98.7	61.7	**95.5**	58.7	94.1	73.2
	Previous MDSA [14, 15]	92.3	96.6	**98.8**	**62.5**	95.4	**59.1**	**94.2**	**73.4**
Proposed self-attention models	Dual-axial SA w/ softmax	92.3	97.1	**98.8**	61.9	95.5	59.3	94.3	73.7
	Text-axial SA	92.7	97.3	**98.8**	63.4	95.8	59.1	94.4	73.7
	Text-axial SA w/o attenuation factor	92.5	97.3	**98.8**	63.2	95.7	59.2	94.4	73.4
	Dual-axial SA	**92.8**	**97.5**	**98.8**	63.5	**95.8**	59.5	94.4	**73.8**

† The results of VDCNN are obtained by selecting the best performance from three different types of pooling.
⋆ The results of SRNN are obtained by the model with default hyperparameters [29].

5.3 Results

We list the experimental results in Table 3. In this table, the results of the previous models except SRNN are obtained from the corresponding references, and the rest of the results are averaged from the results of three tests conducted by us.

Comparisons with other models. Our model is able to achieve beyond state-of-the-art results on five datasets and a little worse results on the rest of three datasets than VDCNN. Strictly speaking, VDCNN is not one but three models, because the results of VDCNN shown in Table 3 are a selection from three variants of VDCNN.

Comparisons with conventional self-attention models. The proposed dual-axial self-attention based model achieves higher accuracies on almost all the datasets by approximately 0.1% to 2% improvement.

Ablation studies.

• Comparing dual-axial self-attention with its softmax-variant: the experimental results strongly support the rationality of the replacement of softmax.

• Comparing dual-axial self-attention with text-axial self-attention: the experimental results show that the introduction of dependencies along the feature axis can improve the performance on the whole.

• Comparing text-axial self-attention models equipped with and without attenuation factor: the experimental results show that the introduction of the attenuation factor is able to slightly improve the performance of the text-axial self-attention.

5.4 Case study

To give an intuitive explanation of the dual-axial self-attention and a comparison between softplus and softmax, we visualize the attention weights of the text-axial and feature-axial self-attention, respectively, as well as their softmax-variants in Figure 5. The attention weights are all trained in a news classification task.

Figure 5 shows the following.

• Both text-axial and feature-axial self-attention are able to ignore unimportant words ("could", "be", "for", "a", etc.) and highlight important words ("Shell", "Oil", "giant", "takeover", etc.).

• The proposed multi-dimensional self-attention is able to obtain distinct attention weights for different dimensions of the same word.

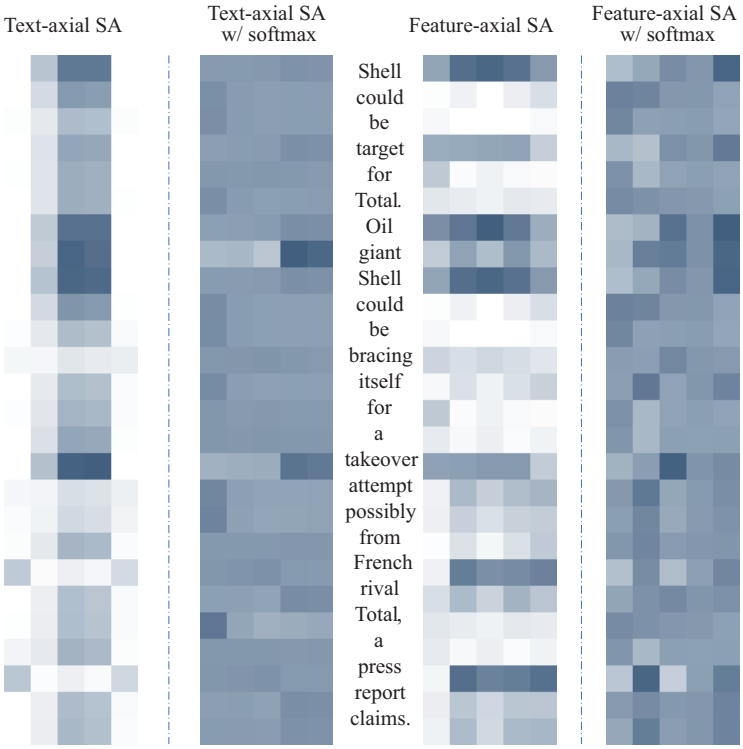

Figure 5 (Color online) Attention visualizations. Each block in this figure represents an attention weight for a specific dimension of an embedded word. For the sake of illustration, the embedding length is set to 5.

• Compared with our proposed model, the attention weights generated by the softmax-variants are less differentiated.

6 Discussion

6.1 Computational complexity

The proposed multi-dimensional self-attention is much faster than the previous ones. For example, it only takes 15 s per epoch to train our proposed model on AG, while it takes more than 260 s for the previous models [14, 15] on the same GPU cards. Note that the dual-axial self-attention includes two self-attention modules. Thus the reduction of the computational complexity to train only one module is even more distinguished compared with the previous models.

The reasons for the acceleration of the proposed multi-dimensional self-attention mainly owing to the following two points. Firstly, the time complexity of our proposed multi-dimensional self-attention is lower than the previous ones. It can figure out that the time complexity of our model is $O(b \times N^2 \times d)$, while that of the previous ones is $O(b \times N^2 \times d^2)$, where b, N and d denote the mini-batch size, the text length, and embedding dimensions, respectively. Secondly, in the previous models, there are several huge 4-dimensional tensors of size $b \times N^2 \times d$ involved in the computation process [30], while in our proposed multi-dimensional self-attention, the maximum tensors are of size $b \times N \times d$. That is to say, the space complexity of our model is lower than the previous ones, making it possible for setting a larger mini-batch size in practice.

6.2 Convergence

Figure 6 demonstrates the learning curves of self-attention models on AG. It can be clearly seen as follows.

• The proposed dual-axial self-attention shows a much lower testing loss than its softmax-variant while they share almost the same training losses. That is, the replacement of softmax improves the generalization ability of our proposed self-attention model.

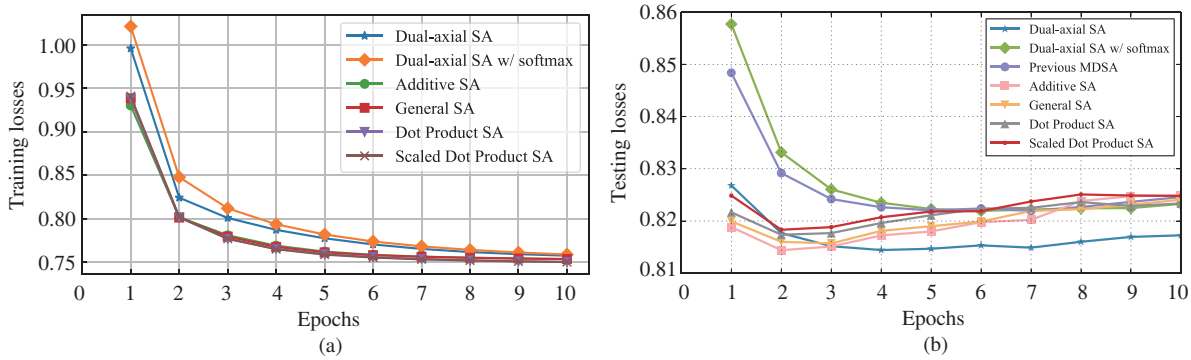

Figure 6 (Color online) Learning curves on AG. (a) and (b) demonstrate training losses and testing losses, respectively.

• The proposed dual-axial self-attention is less likely to overfit and is able to obtain a lower loss on testing set compared with the conventional models.

7 Conclusion

In this paper, we propose a text classification neural network based on a powerful multi-dimensional self-attention model, called dual-axial self-attention. The proposed multi-dimensional self-attention is implemented on the basis of single-dimensional self-attention and thus this method is much more computationally efficient than all the previous ones. The creative idea is that we introduce dependencies along the feature axis as an effective supplement to self-attention. A significant improvement can be obtained by substituting softmax function by softplus function. Furthermore, introducing the attenuation factor is beneficial to the text-axial dependencies modeling. Experiments on eight representative datasets demonstrate the efficacy of our proposed model.

Our research showcases a successful application of feature-axial dependency modeling in the NLP field. For future work, we plan to further explore the effectiveness of feature-axial dependency in other NLP tasks.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61802435, 61802433).

References

1 Bi J W, Liu Y, Fan Z P, et al. Wisdom of crowds: conducting importance-performance analysis (IPA) through online reviews. Tourism Manage, 2019, 70: 460–478
2 Felbo B, Mislove A, Søgaard A, et al. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, 2017. 1615–1625
3 Liu Y, Bi J W, Fan Z P. Ranking products through online reviews: a method based on sentiment analysis technique and intuitionistic fuzzy set theory. Inf Fusion, 2017, 36: 149–161
4 Wang X J, Yu L T, Ren K, et al. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, 2017. 2051–2059
5 Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification. In: Proceedings of Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, Montreal, 2015. 649–657
6 Xiao Y J, Cho K. Efficient character-level document classification by combining convolution and recurrent layers. 2016. ArXiv:1602.00367
7 McCallum A, Nigam K. A comparison of event models for naive Bayes text classification. In: Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, 1998, 752: 41–48
8 Rennie J D M, Shih L, Teevan J, et al. Tackling the poor assumptions of naive Bayes text classifiers. In: Proceedings of the 20th International Conference 2003, Washington, 2003. 616–623
9 Jiang L, Li C, Wang S, et al. Deep feature weighting for naive Bayes and its application to text classification. Eng Appl Artif Intell, 2016, 52: 26–39
10 Jiang L, Wang S, Li C, et al. Structure extended multinomial naive Bayes. Inf Sci, 2016, 329: 346–356
11 Conneau A, Schwenk H, Barrault L, et al. Very deep convolutional networks for natural language processing. 2016. ArXiv:1606.01781
12 Yogatama D, Dyer C, Wang L, et al. Generative and discriminative text classification with recurrent neural networks. 2017. ArXiv:1703.01898
13 Yu Z P, Liu G S. Sliced recurrent neural networks. In: Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018), Santa Fe, 2018. 2953–2964
14 Shen T, Zhou T Y, Long G D, et al. DiSAN: directional self-attention network for RNN/CNN-free language understanding. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial

Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, 2018. 2953–2964
15 Shen T, Zhou T Y, Long G D, et al. Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018), Stockholm, 2018. 4345–4352
16 Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. 2014. ArXiv:1409.0473
17 Luong T, Pham H, Manning C D. Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), Lisbon, 2015. 1412–1421
18 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, 2017. 6000–6010
19 Dhingra B, Liu H X, Yang Z L, et al. Gated-attention readers for text comprehension. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), Vancouver, 2017. 1832–1846
20 Fu J, Liu J, Tian H J, et al. Dual attention network for scene segmentation. 2018. ArXiv:1809.02983
21 Yang B S, Wang L Y, Wong D F, et al. Convolutional self-attention networks. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Minneapolis, 2019. 4040–4045
22 Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018), New Orleans, 2018. 464–468
23 Al-Rfou R, Choe D, Constant N, et al. Character-level language modeling with deeper self-attention. 2018. ArXiv:1808.04444
24 Kingma D P, Ba J. Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, 2015
25 Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010), Sardinia, 2010. 249–256
26 Demsar J. Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res, 2006, 7: 1–30
27 Jiang L, Zhang L, Li C, et al. A correlation-based feature weighting filter for naive Bayes. IEEE Trans Knowl Data Eng, 2019, 31: 201–213
28 Grave E, Mikolov T, Joulin A, 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 (EACL 2017), Valencia, 2017. 427–431
29 Yu Z P. Code of sliced recurrent neural networks (SRNN): zepingyu0512/srnn. 2019. https://github.com/zepingyu0512/srnn
30 Shen T. Code of directional self-attention network (DiSAN): taoshen58/DiSAN. 2019. https://github.com/taoshen58/DiSAN