Dynamic Global-Local Attention Network Based On Capsules for Text Classification

Ji Wang
School of Information and Technology
Zhejiang Sci-Tech University
Hangzhou 310018, China
jiwang2014@outlook.com

Qiaohong Chen*
School of Information and Technology
Zhejiang Sci-Tech University
Hangzhou 310018, China
chen_lisa@zstu.edu.cn

Haolei Pei
School of Information and Technology
Zhejiang Sci-Tech University
Hangzhou 310018, China
haleypei@outlook.com

Qi Sun
School of Information and Technology
Zhejiang Sci-Tech University
Hangzhou 310018, China
sunqi@vip.sina.com

Yubo Jia
School of Information and Technology
Zhejiang Sci-Tech University
Hangzhou 310018, China
jiayubo1964@163.com

***Abstract*—Text classification requires a comprehensive consideration of global and local information for the text. However, most methods only treat the global and local features of the text as two separate parts and ignore the relationship between them. In this paper, we propose a Dynamic Global-Local Attention Network based on Capsules (DGLA) that can use global features to dynamically adjust the importance of local features (e.g., sentence-level features or phrase-level features). The global features of the text are extracted by the capsule network, which can capture the mutual positional relationship of the input features to mine more hidden information. Furthermore, we have designed two global-local attention mechanisms within DGLA to measure the importance of two different local features and effectively leverage the advantages of these two attention mechanisms through the residual network. The performance of the model was evaluated on seven benchmark text classification datasets, and DGLA achieved the highest accuracy on all datasets. Ablation experiments show that the global-local attention mechanism can significantly improve the performance of the model.**

I. Introduction

Text classification is an important research area of natural language processing. Its main research content is to allow computers to understand the content of the text, and then divide it into predefined categories.

With the rise of deep learning, many neural network methods have been introduced into text classification tasks and have achieved good results. However, most of these networks focus on improving the quality of global or local features extracted from text to get a better text representation, which greatly increases the complexity of the model. The global features here are the model's understanding of the entire text, while the local features represent sentence-level or phrase-level features extracted from the text. Moreover, it is not comprehensive enough to use only global or local features of the text to represent the entire text, as they reflect information at different levels of the text. Some hybrid models can obtain a high-quality text representation by concatenating the global and local features of the text, but they treat the global and local features as two separate parts and ignore the relationship between them.

To solve the above shortcomings, we propose a Dynamic Global-Local Attention Network based on Capsules (DGLA), which can explore the relationship between global and local features of text by simulating people's reading habits. For an incomprehensible text, people may not understand its content well after the first reading, but they can get some initial global understanding of the entire text. When people read the text again, these initial global understandings will guide them to pay more attention to some important sentences or phrases, which can deepen their understanding of the text. To simulate this reading habit, DGLA introduced the capsule network to obtain multiple global features from the text and then used them to measure the importance of local features. Furthermore, we have designed two global-local attention mechanisms within DGLA to measure the importance of two different local features, called Global-Local Attention A (GLA-A) and Global-Local Attention B (GLA-B). GLA-A uses global features extracted from sentence-level features by capsule networks to measure the importance of phrase-level features. GLA-B uses global features extracted from phrase-level features by capsule networks to measure the importance of sentence-level features. And DGLA effectively leverage the advantages of these two attention mechanisms through the residual network. We evaluated the performance of our model on seven benchmark text classification datasets and achieved the highest accuracy on all datasets. Ablation experiments show that the global-local attention mechanism can significantly improve the performance of the model. The main contributions of this paper are listed as follows:

- DGLA introduced the capsule network to extract global features from the text. The dynamic routing process

*Corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

inside the capsule network can dynamically adjust the attention weight of the input features to mine more hidden information.

- We have designed a global-local attention mechanism that can dynamically adjust the importance of local features using the global feature.
- We use two global-local attention mechanisms within DGLA to measure the importance of two different local features and effectively leverage the advantages of these two attention mechanisms through the residual network.

II. Related Work

Text classification is an important research area of natural language processing. Based on the assumption that the meaning of a word depends on its context, Mikolov et al. [1] proposed a method of converting words into word vectors, which broke the barrier between natural language processing and deep learning. Since then, many deep learning methods have been introduced into text classification tasks. Convolutional neural networks (CNN) can effectively extract word or phrase features from sentences. Kim [2] uses three one-dimensional convolutional neural networks with n-gram sizes of 3, 4, and 5 to extract important local features from the text. This method proves the feasibility of CNN in the text classification task. Conneau et al. [3] introduce deep convolutional neural networks that excel in computer vision into text classification. The results show that within a certain depth range, the performance of the model improves with increasing depth. Long Short-Term Memory (LSTM) [4] can effectively learn the long-term dependence of sequences and solve the problems of gradient disappearance and gradient explosion existing in traditional recurrent neural networks. Unlike the conventional method, which uses the last hidden state of LSTM as its output, Lai et al. [5] use a one-dimensional (1D) max-pooling layer to obtain important semantic features, which significantly improves the accuracy of text classification. Applying the one-dimensional max-pooling operation to the output of the LSTM may destroy the positional relationships between semantic features. Zhou et al. [6] use two-dimensional (2D) convolutional neural networks and 2D max-pooling operations to sample important semantic features while retaining their positional relationships. This operation significantly improves the performance of the model. As the attention mechanism can obtain the dependencies between long-distance words or phrases. Some methods have achieved better performance by combining deep neural networks and attention mechanisms on text classification tasks. Yang et al. [7] applied word-level and sentence-level attention mechanisms to build document representations that can capture important word and sentence information. Vaswani et al. [8] proposed a feature extraction method based on the attention mechanism. This extractor not only extracts information better than LSTM, but also has faster calculation speed, and has been widely used in recent pre-trained models [9], [10]. However, most of the above work focused on improving the quality of the final global or local features to get a better text representation, which greatly increased the complexity of the model. Moreover, using only global or local features to represent the entire text is not comprehensive enough because they reflect different levels of information in the text. Some hybrid models can obtain a high-quality text representation by concatenating the global and local features of the text, but they treat the global and local features of the text as two separate parts and ignore the relationship between them.

Capsule network [11] can extract richer information by learning the correlation between input features. Zhao et al. [12] designed three improved dynamic routing methods to reduce the interference of noise information on the dynamic routing process. Wang et al. [13] designed a capsule model using the attention mechanism to mimic the dynamic routing process of the original capsule, which effectively speeds up the calculation of each capsule unit and achieve the state-of-art performance. Yoon et al. [14] introduced a self-attention mechanism inside the capsule to improve its feature extraction capabilities, and then stitched the output of each capsule to obtain a high-quality instance representation. These work not only proved the feasibility of capsule network in text classification, but also demonstrated its excellent feature extraction capabilities.

Compared with existing models, our capsule-based dynamic global-local attention network introduces the capsule network to obtain global features from the text. The dynamic routing process in the capsule network can adjust the attention weight of the input features to mine more hidden information. Moreover, our model dynamically adjusts the importance of local features through global features, rather than directly concatenating them together. In this way, the final text representation contains not only global and local information of the text, but also the relationship between them.

III. Dynamic Global-Local Attention Network based on Capsules

This paper proposes a Dynamic Global-Local Attention Network based on Capsules (DGLA) that can use global features to dynamically adjust the importance of local features. The architecture of DGLA is shown in Fig. 1, It contains five modules: feature extraction module, capsule module, dynamic global-local attention module, residual module, and classification module.

DGLA first uses the feature extraction module to extract the sentence-level and phrase-level features from the text. Next, the capsule neural network is introduced in the capsule module to obtain global features based on sentence-level and phrase-level features, respectively. Then, in the dynamic global-local attention module, we designed two two attention to dynamically measure the importance of two different local features, called Global-Local Attention A (GLA-A) and Global-Local Attention B (GLA-B). GLA-A uses global features to measure the importance of phrase-level features, while GLA-B uses global features to measure the importance of sentence-level features. The Residuals module combines the results of the capsule module and the dynamic global-local attention module

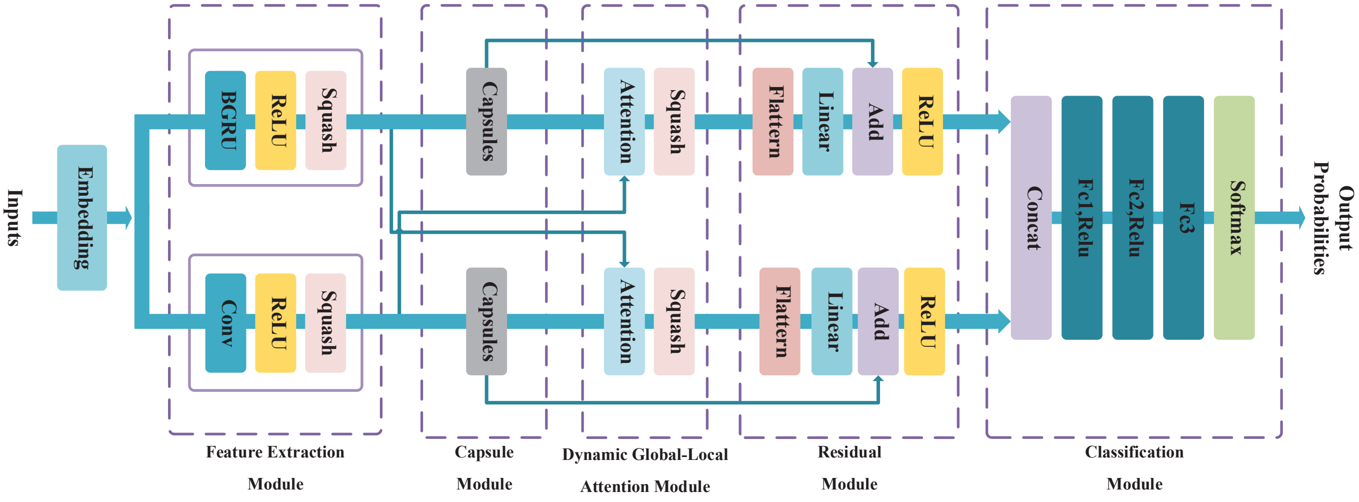

Fig. 1. Architecture of Dynamic Global-Local Attention Network based on Capsules.

using the residual network. In this way, when DGLA merges these two types of global-local attention mechanisms, it can automatically adjust the contributions of GLA-A and GLA-B to the model, thereby effectively leverage their advantages. Finally, the classification module outputs the probability of each text category. The following sections describe the details of these modules.

A. Feature Extraction Module

In this module, Bidirectional Gated Recurrent Unit (BGRU) and N-gram Convolutional Neural Network (CNN) are applied to extract the semantic features (sentence-level features) and n-gram features (phrase-level features) from the text, respectively.

a) N-gram Convolutional Neural Network: The goal of this layer is to extract n-gram features (phrase-level features) from text.

Given a text of length L, the first thing we need to do is convert it to a word embedding matrix $X = [x_1, \ldots, x_i, \ldots x_L] \in \mathbb{R}^{L \times V}$. Let $W^c \in \mathbb{R}^{k \times V}$ be one of the filters in N-gram CNN, where V is the dimension of word embedding and k is the window size of the convolution operation. The filter W^c convolves each possible word window $x_{i:i+k-1}$ with the stide of 1 to extract all n-gram features $m^c \in \mathbb{R}^{L-k+1}$ from text:

$$\begin{gathered} m^c = Concat(m_1^c, ..., m_i^c, ..., m_{L-k+1}^c) \\ where\ m_i^c = f(x_{i:i+k-1} \otimes W^c + b_0) \end{gathered} \tag{1}$$

Where $m_i^c \in \mathbb{R}$ is one of the n-gram feature extracted from the phrase, f is the activate function and $\otimes$ is the convolution operation, which first performs element-wise multiplication and then sums the results. Now, we have described the process of using a filter to extract all n-gram features from the text. Next, we use the B filters with the same word window size to perform the above convolution operations on the text and rearrange their results to get the final n-gram feature matrix:

$$M = [m^1, ..., m^c, \ldots, m^B] \in \mathbb{R}^{(L-k+1) \times B} \tag{2}$$

b) Bidirectional Gated Recurrent Unit: Long Short-Term Memory (LSTM) [4] is a widely used neural network in natural language processing. Compared with traditional recurrent neural networks (RNN), LSTM can learn the long-term dependencies between words more effectively and solve the problem of gradient disappearance, but its operation speed is very slow. GRU (Gated Recurrent Unit) [15] simplifies the gating mechanism of LSTM and accelerates the training speed while maintaining its performance.

Just like Bidirectional Long Short-Term Memory [16], GRU can also be bidirectional. The hidden state of each unit in the Bidirectional Gated Recurrent Unit (BGRU) contains not only its past information, but also its future information. This is achieved by two GRUs processing text in different directions, one from left to right to obtain the hidden state containing past information, and the other from right to left to obtain the hidden state containing future information:

$$\overrightarrow{h}_t = GRU(\overrightarrow{h}_{t-1}, \overrightarrow{x}_t) \tag{3}$$

$$\overleftarrow{h}_t = GRU(\overleftarrow{h}_{t-1}, \overleftarrow{x}_t) \tag{4}$$

$$h_t = Concat(\overrightarrow{h}_t, \overleftarrow{h}_t) \tag{5}$$

In this layer, we use BGRU to quickly obtain the semantic features of the text. Hence, given the word embedding matrix $X = [x_1, \ldots, x_i, \ldots x_L] \in \mathbb{R}^{L \times V}$, we can obtain L semantic features:

$$H = [h_1, \ldots, h_t, \ldots, h_L] \in \mathbb{R}^{L \times 2d_h} \tag{6}$$

Where d_h is the hidden size of GRU, and L is the length of the text.

c) Squash Function: Given n-gram features $M \in \mathbb{R}^{(L-K+1) \times B}$ and semantic features $H \in \mathbb{R}^{L \times 2d_h}$, the *squash* function was adopted to normalize these features. The function's operation on a single feature is shown below:

$$v_{out} = \frac{||v_{in}||^2}{1 + ||v_{in}||^2} \frac{v_{in}}{||v_{in}||} \tag{7}$$

After normalizing each feature in M and H using the $squash$ function, we get two squashed vector matrices:

$$V^c = [v_1^c, ..., v_i^c, ...v_{L-K+1}^c] \in \mathbb{R}^{(L-K+1)\times B} \tag{8}$$

$$V^l = [v_1^l, ..., v_i^l, ...v_L^l] \in \mathbb{R}^{L\times 2d_h} \tag{9}$$

Where V^c is the normalized output of n-gram features M, and V^l is the normalized output of semantic features H.

B. Capsule Module

In this module, we introduce the capsule network [11] to extract global features. The dynamic routing process inside the capsule network can adjust the attention weight of input features to mine more hidden information. We combine the capsule network with N-gram CNN and BGRU to obtain the global features based on semantic features and n-gram features, respectively.

The input of the capsule network is v_i^t. We use v_i^c and v_i^l to represents i-th vector of V^c and V^l respectively. Here V^l and V^c are the output vectors of the feature extraction module. The calculation process of the capsule module is as follows:

$$\hat{u}_{j|i}^t = W_{ij}^t v_i^t \tag{10}$$

$$s_j^t = \sum_i c_{ij}\hat{u}_{j|i}^t \tag{11}$$

$$c_{ij} = \frac{exp(b_{ij})}{\sum_k exp(b_{ik})} \tag{12}$$

Where c_{ij} is the coupling coefficient, which is updated through the dynamic routing process. Weight matrix W_{ij}^t transforms input features from input space to output space. s_j^t is a global feature based on all input features. The coupling coefficients between global feature s_j^t and all the input features sum to 1 and are determined by a "routing softmax" with b_{ij} initialized to 0. Then, the $squash$ function, which shown in (7), is used to scale the globally represented modulus length between 0 and 1:

$$g_j^t = squash(s_j^t) \tag{13}$$

The dynamic routing process is computed by following Algorithm 1:

Algorithm 1 Dynamic Routing Algorithm

```
procedure ROUTING($\hat{u}_{j|i}$ , $r$ )
  for all input feature $i$ in input layer:
  for all global feature $j$ in output layer: $b_{ij}$=0
    for $r$ iterations do
        $c_{ij} = softmax(b_{ij})$
        $s_j^t = \sum_i c_{ij}\hat{u}_{j|i}^t$
        $g_j^t = squash(s_j^t)$
        $b_{ij} = b_{ij} + \hat{u}_{j|i}^t g_j^t$
    return $g_j^t$
```

Where r (default 3) is the number of iterations, and $g_j^t \in \mathbb{R}^{d_c}$ is one of the global features based on the input features. Now we have described the process of generating a global feature based on all input features. Therefore, for $j = 1, ..., n_c$ and all v_i^t in V^c or V^l, we can generate two global representations, respectively.

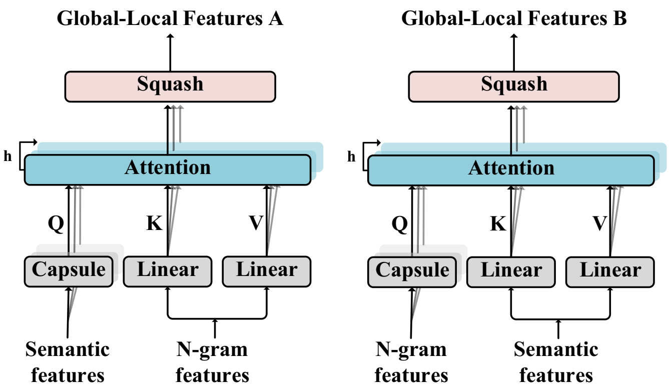

Fig. 2. Architecture of global-local attention A (left) and global-local attention B (right).

$$Global_c = [g_1^c, g_2^c, \ldots, g_{n_c}^c] \in \mathbb{R}^{n_c\times d_c} \tag{14}$$

$$Global_l = [g_1^l, g_2^l, \ldots, g_{n_c}^l] \in \mathbb{R}^{n_c\times d_c} \tag{15}$$

Where $Global_c$, $Global_l$ is the global features corresponding to V^c and V^l, n_c is the number of global features, and d_c is the dimension of a single global feature.

C. Dynamic Global-Local Attention Module

In this module, we use the global feature to dynamic measure the importance of semantic features and n-gram features, respectively. To achieve this, we designed two attention mechanisms, called Global-Local Attention A (GLA-A) and Global-Local Attention B (GLA-B), as shown in Fig. 2. GLA-A uses global features extracted from semantic-level features by capsule networks to measure the importance of n-gram features. GLA-B uses global features extracted from n-gram features by capsule networks to measure the importance of semantic features. The following sections provide details of these two attention mechanisms.

a) Global-Local Attention A: We design the global-local attention mechanism inspired by [8]. The GLA-A can be described as a fuzzy search of all key-value pairs for a given query, where the query is a global feature based on semantic features, keys and values are obtained by the linear transformation of n-gram features. The output of this attention mechanism is a weighted sum of values, and the weight of each value is determined by the similarity between the query and the key corresponding to the value.

In practice, we simultaneously calculate the importance of n-gram features on a set of global features and pack them together to obtain a matrix Q. The keys and values obtained by the linear transformation of the n-gram features are also packed together to obtain the matrices K and V, respectively. The output matrix of this attention mechanism is calculated as:

$$Attention(Q, K, V) = softmax(QK^T)V \tag{16}$$

The specific calculation details of GLA-A are as follows:

$$Att_{lc} = Attention(Global_l, V^cW^{KC}, V^cW^{VC}) \tag{17}$$

Where $W^{KC} \in \mathbb{R}^{2d_h \times d_c}$, $W^{VC} \in \mathbb{R}^{2d_h \times d_c}$, d_h is the hidden size of a single GRU in the BGRU. d_c is the dimension of the global feature.

Then, the non-linear activation squash function, which is shown in (7), is adopted to make the magnitude of the output consistent with the global features.

$$Att_c = squash(Att_{lc}) \tag{18}$$

b) Global-Local Attention B: Different from GLA-A, in GLA-B we use the global feature based on n-gram features to help us measure the importance of semantic features. The calculation details of this attention mechanism is shown below:

$$Att_{cl} = Attention(Global_c, V^lW^{KL}, V^lW^{VL}) \tag{19}$$

$$Att_l = squash(Att_{cl}) \tag{20}$$

Where $W^{KL} \in \mathbb{R}^{B \times d_c}$, $W^{VL} \in \mathbb{R}^{B \times d_c}$, B is the number of convolution filters in N-gram CNN.

D. Residual Module

The Residuals module combines the results of the capsule module and the dynamic global-local attention module using the residual network. In this way, when DGLA merges these two types of global-local attention mechanisms, it can automatically adjust the contributions of GLA-A and GLA-B to the model, thereby effectively leverage their advantages. The calculation details are as follows:

$$Glo_c = Flattern(Global_c)W^C \tag{21}$$

$$Glo_l = Flattern(Global_l)W^L \tag{22}$$

$$Att_A = max(0, Flattern(Att_c)W^{CO} + Glo_l) \tag{23}$$

$$Att_B = max(0, Flattern(Att_l)W^{LO} + Glo_c) \tag{24}$$

Where $W^{CO} \in \mathbb{R}^{n_c d_c \times d_o}$, $W^C \in \mathbb{R}^{n_c d_c \times d_o}$, $W^{LO} \in \mathbb{R}^{n_c d_c \times d_o}$, $W^L \in \mathbb{R}^{n_c d_c \times d_o}$,$d_o$ is the dimension of the text representation. And the $Flattern()$ function is used to expand multidimensional feature into one-dimensional feature. Att_A and Att_B are the outputs of Global-Local Attention A and Global-Local Attention B, respectively.

E. Classification module

In this module, we concatenate the output of two global-local attention mechanisms to get the final representation of the text:

$$V^{final} = Concat(Att_A, Att_B) \tag{25}$$

To prevent overfitting, we adopt dropout operation on V^{final} before inputting it into the fully connected layer and the dropout rate is set to 0.3. Next, we use three fully connected layers for further feature extraction. The first two fully connected layers use $ReLU$ as the activation function, and the last one uses $softmax$ to output the probability distribution for each text category.

TABLE I
DISTRIBUTION OF SEVEN BENCHMARK DATASETS.

	Train	Dev	Test	Classes	Classification Task
MR	8.6k	0.9k	1.1k	2	review classification
SST-1	8.5k	1.1k	2.2k	5	sentiment analysis
SST-2	6.9k	0.8k	1.8k	2	sentiment analysis
SUBJ	8.1k	0.9k	1.0k	2	opinion classification
TREC	5.4k	0.5k	0.5k	6	question categorization
CR	3.1k	0.3k	0.4k	2	review classification
AG's news	108k	12.0k	7.6k	4	news categorization

IV. EXPERIMENT AND RESULTS

A. Datasets

In this paper, we evaluate the performance of our model on seven data sets that are widely used for text classification. The details of these data sets are as follows:

- **Movie Review (MR)** [17] is a collection of movie reviews in English. Each sentence in this dataset is marked as positive or negative, with a total of 5331 positive sentences and 5331 negative sentences.
- **Stanford Sentiment Treebank(SST-1)** [18] is an extension of MR, which provides training, validation, and test data sets, and more fine-grained labeling of the text (very positive, positive, neutral, negative, very negative). The data distribution for each sentiment category is 1837, 3118, 2237, 3147, 1516.
- **SST-2** [18] is the same as SST-1, but removed the neutral reviews and reduced its number of categories to positive and negative. The data distribution is 4955, 4663.
- **Subjectivity dataset (SUBJ)** [19] Sentences in this data set are labeled as subjective and objective, and the distributions of subjective and objective sentences are 5000 and 5000, respectively.
- **TREC** question dataset [20], the task is to divide a question into 6 categories (about number information, location, people, etc.).
- **Customer reviews (CR)** [21] dataset contains user reviews for various products. This dataset marks reviews as positive or negative and contains a total of 2411 positive reviews and 1373 negative reviews.
- **AG's news (AG's)** [22] contains 496,835 news articles from more than 2000 news sources in 4 major categories of the AG News Corpus. Each category has 30,000 training samples and 1900 test samples.

TABLE I shows the details of these datasets and the partitioning of the training/validation/test dataset during the experiment.

B. Baseline Method

We evaluate the performance of the proposed model on seven benchmark datasets and compare it with the results of other baseline methods on these datasets. The details of these baseline methods are as follows:

- **LSTM** [4]: This is s a widely used neural network for natural language processing, which can not only learn long-term dependencies in text, but also solve the problem of vanishing gradients.

- **Bi-LSTM** [16]: Bi-LSTM uses two LSTMs to process text from two different directions so that the output corresponding to each word contains its past and future information.
- **Tree-LSTM** [23]: extends LSTM to a tree-type input structure..
- **LR-LSTM** [24]: LR-LSTM use linguistically regularized to extend LSTM.
- **CNN-rand** [2]: A model uses three different n-gram size convolutional network to extract local information, and its word embedding matrix is randomly initialized and participates in training.
- **CNN-static** [2]:Same as CNN-rand, but its word embedding matrix is initialized with pre-trained word vectors and does not participate in training.
- **CNN-non-static** [2]: Same as CNN-rand, but its word embedding matrix is initialized with pre-trained word vectors and participate in training.
- **CL-CNN** [25]: A convolutional network whose convolution operations are calculated at the character level.
- **VD-CNN** [3]: A very deep convolutional neural network model that uses one-dimensional convolution and one-dimensional pooling internally.
- **Capsule-A** [12]: A model proposes three strategies for improving the dynamic routing process of capsule neural networks, and uses a parallel network with n-gram size 3 in the convolutional layer.
- **Capsule-B** [12]: Same as Capsule-A, but it uses three parallel networks with n-gram sizes of 3, 4, and 5 in the convolutional layer.

C. Implemention Details

In our experiment, the embedding layer is initialized by word vectors pre-trained with Glove [26], and its dimension is 300. The xavier normal distribution [27] is used to initialize all out-of-vocabulary words. The length L of the text is fixed at 50. The dimension of the output vector of the the N-Gram CNN layer and BGRU layer is set to 100. The N-gram size of CNN is set to 3. When using the capsule network to extract the global representation, the capsule's number n_c is equal to the number of categories, and the dimension d_c of the output vector is set to 64. The global-local representation size d_o is set to 100 and the dropout rate is set to 0.3.

We use cross-entropy as the loss function of the model and use L2 regularization to prevent overfitting, and the regularization coefficient is 0.0001. The model parameters were trained using Adam [28] with the learning rate of $1e-3$, and the parameters of the model were initialized using the xavier normal distribution.

D. Model Variations

We also designed several model variants to perform ablation experiments on DGLA to prove the importance of the global-local attention mechanism. The information for these variants is as follows:

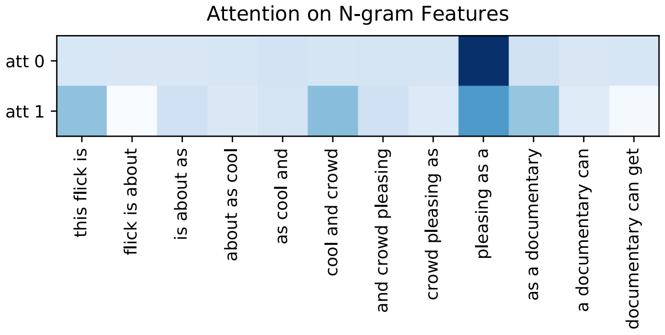

Fig. 3. Visualizing global-local attention A.

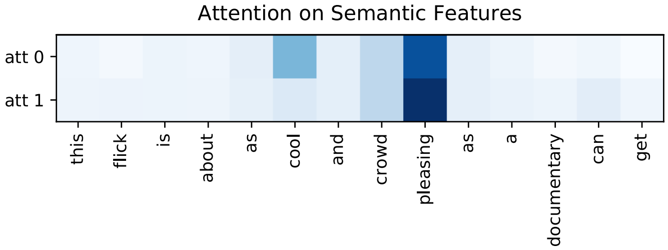

Fig. 4. Visualizing global-local attention B.

- **CNN + BGRU**: A model only extracts phrase-level and sentence-level information.
- **CNN + BGRU + Capsule**: A model only uses global representation extracted on phrase-level and sentence-level information.
- **CNN + BGRU + Attention-A**: Removed the global-local attention B module in DGLA, other modules remain unchanged.
- **CNN + BGRU + Attention-B**: Removed the global-local attention A module in DGLA, other modules remain unchanged.

E. Experimental Results

We evaluate the performance of our model by its classification accuracy on the seven benchmark text classification datasets. TABLE II shows the classification accuracy of Dynamic Global-Local Attention Network based on Capsules (DGLA) and other baseline methods on these benchmark datasets. Compared to the baseline method, our model improves accuracy by a maximum of 2.6% and a minimum of 0.2%. And the accuracy of DGLA is on average 1.3% higher than all baseline methods. The experimental results of TABLE II show that the performance of DGLA is better than all baseline methods.

In TABLE III, we show the results of the ablation experiments of DGLA. In this table, it can be observed that CNN + BRUG + Capsule model has higher performance than CNN + BGRU model. And CNN + BGRU + Attention-B

TABLE II
CLASSIFICATION ACCURACY RATE (%) ON SEVERAL STANDARD BENCHMARKS

Model	MR	SST-2	SUBJ	TREC	CR	SST-1	AG's
LSTM	75.9	80.6	89.3	86.8	78.4	45.6	86.1
Bi-LSTM	79.3	83.2	90.5	89.6	82.1	46.5	88.2
Tree-LSTM	80.7	85.7	91.3	91.8	83.2	48.1	90.1
LR-LSTM	81.5	87.5	89.9	-	82.5	48.2	-
CNN-rand	76.1	82.7	89.6	91.2	79.8	45.0	92.2
CNN-static	81.0	86.8	93.0	92.8	84.7	45.5	91.4
CNN-non-static	81.5	87.2	93.4	93.6	84.3	48.0	92.3
CL-CNN	-	-	88.4	85.7	-	-	92.3
VD-CNN	-	-	88.2	85.4	-	-	91.3
Capsule-A	81.3	86.4	93.3	91.8	83.8	-	91.8
Capsule-B	82.3	86.8	93.8	92.8	85.1	-	92.8
DGLA	**83.3**	**89.0**	**94.9**	**94.2**	**85.3**	**50.6**	**93.4**

TABLE III
CLASSIFICATION ACCURACY RATE (%) OF DGLA AND ITS VARIANTS

Model	MR	SST-2	SUBJ	TREC	CR	SST-1	AG's
CNN+BGRU	82.0	86.1	93.5	91.6	84.1	46.3	92.2
CNN + BGRU + Capsule	82.0	86.2	93.7	92.0	84.5	47.1	92.4
CNN + BGRU + Attention-A	82.5	86.6	94.2	**94.6**	84.5	49.1	93.1
CNN + BGRU + Attention-B	83.1	87.8	93.8	92.0	84.0	48.0	92.5
DGLA	**83.3**	**89.0**	**94.9**	94.2	**85.3**	**50.6**	**93.4**

performs well on MR and SST-2, CNN + BGRU + Attention-A performs well on the remaining datasets. Comparing DGLA with its variants, DGLA performs better on most datasets than its variants except TREC. And DGLA achieves a maximum improvement in accuracy of 1.5% on the SST-1 dataset.

We also randomly selected a sample from the MR dataset and visualized its global-local attention A and global-local attention B to observe which semantic features and n-gram features DGLA paid more attention to. Fig. 3 and Fig. 4 show the visualization of these two attention mechanisms, respectively. Fig. 3 visualize global-local attention A, and we can observe that the model puts more attention on important n-gram features such as ”cool and crowd” and ”pleasing as a”. Fig. 4 visualize the global-local attention B, which also successfully focuses on important semantic features such as ”cool” and ”pleasing”.

V. DISCUSSION

The experimental results in TABLE II show that DGLA performs better than other baseline methods on text classification tasks. From the results of the ablation experiments in TABLE III, we can find that the CNN + BGRU + Cpausle model has obvious advantages over the CNN + BGRU model in most data sets, which proves that using the capsule network can indeed extract more hidden information from the text. The CNN + BGRU + Attention-A model and the CNN + BGRU + Attention-B model only have one more global-local attention mechanism than the CNN + BGRU + Capsule model, but the results in TABLE III show that their performance has been significantly improved. This is because the global local attention mechanism can use the global feature to dynamically adjust the attention weight of local features to help the model find more important local features, rather than simply connecting them together. In this way, the final text representation contains not only global and local information of the text, but also the relationship between them. From TABLE III, we can also find that the CNN + BGRU + Attention-B model performs well on MR and SST-2 datasets, and the CNN + BGRU + Attention-A model performs well on the remaining datasets, which indicates that each of these two global-local attention mechanisms has advantages and disadvantages in different classification tasks. DGLA combines these two global-local attention mechanisms and its performance has been significantly improved on most data sets. This is because DGLA's residual module can automatically learn the importance of these two global-local attention mechanisms according to different classification tasks to effectively leverage their advantages. The accuracy of DGLA on the TREC dataset is 0.4% lower than the CNN + BGRU + Attention-A model. One possible reason for this result is that the performance of the CNN + BGRU + Attention-B model is much lower than the performance of the CNN + BGRU + Attention-A model, so when DGLA merges them, the performance will decrease. And the visualization of the attention in Fig. 3 and Fig. 4 further proves that the global features can indeed help us find important local features.

VI. CONCLUSION

We propose a Dynamic Global-Local Attention Network based on Capsules (DGLA) that can use global features to dynamically adjust the importance of local features. DGLA introduced the capsule network to obtain multiple global features from the text and then used them to measure the importance of local features. Furthermore, we have designed two global-local attention mechanisms within DGLA to measure the importance of two different local features and effectively leverage the advantages of these two attention mechanisms through the residual network. We evaluated the performance of our model on seven benchmark text classification datasets, and DGLA achieved the highest accuracy on all datasets. Ablation experiments show that the global-local attention mechanism can significantly improve the performance of the model.

In the future, we will apply this global-local attention mechanism to other classification tasks, such as aspect-based opinion classification, sentiment analysis, and multi-label text classification. Furthermore, we will extract more different types of features from the text and design more global-local attention mechanisms to further explore the relationship between global and local features of the text.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under grant number 51775513.

References

[1] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[2] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.

[3] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, “Very deep convolutional networks for natural language processing,” *arXiv preprint arXiv:1606.01781*, vol. 2, 2016.

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

[5] S. Lai, L. Xu, K. Liu, and J. Zhao, “Recurrent convolutional neural networks for text classification,” in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[6] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, “Text classification improved by integrating bidirectional lstm with two-dimensional max pooling,” *arXiv preprint arXiv:1611.06639*, 2016.

[7] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, “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*, 2016, pp. 1480–1489.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[9] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” *arXiv preprint arXiv:1906.08237*, 2019.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.

[11] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” in *Advances in neural information processing systems*, 2017, pp. 3856–3866.

[12] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, and Z. Zhao, “Investigating capsule networks with dynamic routing for text classification,” *arXiv preprint arXiv:1804.00538*, 2018.

[13] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, “Sentiment analysis by capsules,” in *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 2018, pp. 1165–1174.

[14] D. Yoon, D. Lee, and S. Lee, “Dynamic self-attention: Computing attention over words dynamically for sentence embedding,” *arXiv preprint arXiv:1808.07383*, 2018.

[15] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[16] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[17] B. Pang and L. Lee, “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales,” in *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2005, pp. 115–124.

[18] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, “Parsing with compositional vector grammars,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 455–465.

[19] B. Pang and L. Lee, “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts,” in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 271.

[20] X. Li and D. Roth, “Learning question classifiers,” in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2002, pp. 1–7.

[21] M. Hu and B. Liu, “Mining and summarizing customer reviews,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

[22] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in *Advances in neural information processing systems*, 2015, pp. 649–657.

[23] K. S. Tai, R. Socher, and C. D. Manning, “Improved semantic representations from tree-structured long short-term memory networks,” *arXiv preprint arXiv:1503.00075*, 2015.

[24] Q. Qian, M. Huang, J. Lei, and X. Zhu, “Linguistically regularized lstms for sentiment classification,” *arXiv preprint arXiv:1611.03949*, 2016.

[25] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in *Advances in neural information processing systems*, 2015, pp. 649–657.

[26] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[27] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[28] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.