Fuzzy Systems and Data Mining VII
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA210185

Multi-Granularity and Internal-External Correlation Residual Model for Chinese Sentence Semantic Matching

Lan Zhang and Hongmei Chen [1]
School of Information Science and Engineering, Yunnan University
Kunming 650091, China

Abstract. Sentence semantic matching (SSM) is central to many natural language processing tasks. This is especially the case for Chinese sentence semantic matching due to the complexity of the semantics, missing semantics and semantic confusion are more likely to occur. Existing methods have used enhanced text representations and multiple matching strategies to address these problems but there is still great potential to capture deep semantic information for Chinese text. This paper proposes a Multi-Granularity and Internal-External correlation Residual model (MGIER) to better capture the deep semantic information and to alleviate the missing semantics effectively. First, the MGIER model utilizes character/word granularity to capture fine-grained information. Then, soft alignment attention is employed to enhance the correlation between characters/words in a sentence, called internal correlation, and the correlation between sentences, called external correlation. In particular, this method uses residual connections to preserve more semantic information from the bottom embedding layer to the top prediction layer. Experimental results show that the proposed method achieves state-of-the-art performance for Chinese SSM, and, compared with pre-trained models, the method also achieves better performance with fewer parameters.

Keywords. Chinese text matching, sentence semantic matching, multi-granularity, residual encoding, soft alignment attention, BiLSTM

1. Introduction

Sentence semantic matching (SSM), which is used to identify whether or not two sentences have the same meaning, is widely used in various applications, such as intelligent customer service, information retrieval, and plagiarism detection.

Due to the complexity of the semantics in the Chinese language, missing semantics and semantic confusion are more likely to occur. With the development of deep learning, such as the attention mechanism [1, 2] and Siamese networks [3], it is possible to capture deep semantic information of sentences. In English SSM tasks, [4, 5] lead the way in using multi-granularity to extract fine-grained information. Although the character-granularity is beneficial to enrich English text representation, one single English character does not express meaning. However, a Chinese character is able to represent a definite meaning, which can convey more semantic information. Thus, [6-8]

[1] Corresponding Author, Hongmei Chen, School of Information Science and Engineering, Yunnan University, Kunming 650091, China; Email: hmchen@ynu.edu.cn

utilize character/word granularity to capture semantic information from Chinese text and achieve remarkable performance. In order to capture more semantic information, [2] employs the residual connections to preserve information. [9] preserves semantic information from the bottom layer to the topmost layer. Although existing methods capture deep semantic information from different perspectives, they cannot completely overcome the missing semantics problem.

Motivated by existing methods, we propose a Chinese SSM model, named the Multi-Granularity and Internal-External correlation Residual model (MGIER). We not only alleviate the problem of missing semantics in the encoding process, but also in the information propagation process. Moreover, we capture more sentence correlation information based on multi-granularity. For comparison, we list the core components of related methods in Table 1.

Table 1. Core components of related methods.

Method	Multi-Granularity	Siamese	Residual Connection	Internal correlation	External correlation	From bottom to top	Multi-Residual
MGIER	√	√	√	√	√	√	√
ICE [8]	√	√	×	×	√	×	×
MGF [7]	√	√	×	×	×	×	×
DRCN [9]	×	√	×	×	√	√	×
ESIM [1]	×	√	×	×	√	×	×

In Table 1, the term "Siamese" refers to encoding information by the same network, which is detailed in section 2. "From bottom to top" refers to preserving information in every layer of the models. "Multi-Residual" refers to using residual connections in multiple layers of models.

In summary, we mainly make the following contributions in this work:

- We capture the semantic information of Chinese sentences from the bottom embedding layer to the top prediction layer by using the residual connections.
- We utilize character/word granularity to capture fine-grained information of sentences, and employ soft alignment attention to enhance the internal correlation and the external correlation.

2. Related Work

Earlier SSM methods mainly focused on text representation [3, 10-12]. These obtained the vector representation for two sentences separately at first, and then employed distance measures to compute the semantic similarity of the two sentences. Among them, the Siamese network is still used today because the feature of sharing parameters makes the model smaller and easier to train. A key issue for such models is that there is no explicit interaction between the two sentences.

To enhance interaction of sentences, some methods incorporate attention mechanism into the SSM task [1, 13]. ESIM [1] employs soft alignment attention to enhance the correlation between two sentences. Inspired by ESIM, some methods introduce many complex matching strategies to enhance the correlation between two sentences [2, 5, 13-14]. For these methods, missing semantics is still a difficult issue. To address this problem, researchers discovered that the granularity of text is also crucial for capturing semantic information [4-8], especially for Chinese SSM. In

particular, the MGF model [7] integrates character/word granularity and achieves remarkable results. However, this work does not consider the correlation of sentences. Thus, ICE [8] employs soft alignment attention to extract correlation information, but only considering the internal correlation of sentences is far from adequate. Some methods use residual connections [2] and dense connections [9] to capture semantic information for SSM. However, they do not incorporate multi-granularity and capture sufficient correlation information.

Recently, pre-trained language models (PTMs) have achieved great success in various natural language processing tasks, due to their capacity to capture deep contextualized information [15-19]. Although BERT [15] and NEZHA [17] achieve remarkable results for Chinese SSM, they need to be trained on high-quality datasets and their migration ability is also very limited.

Different from existing methods, we propose the MGIER model to capture more and deeper semantic information. With less computational cost, our method achieves better performance compared with the state-of-the-art methods.

3. MGIER Model

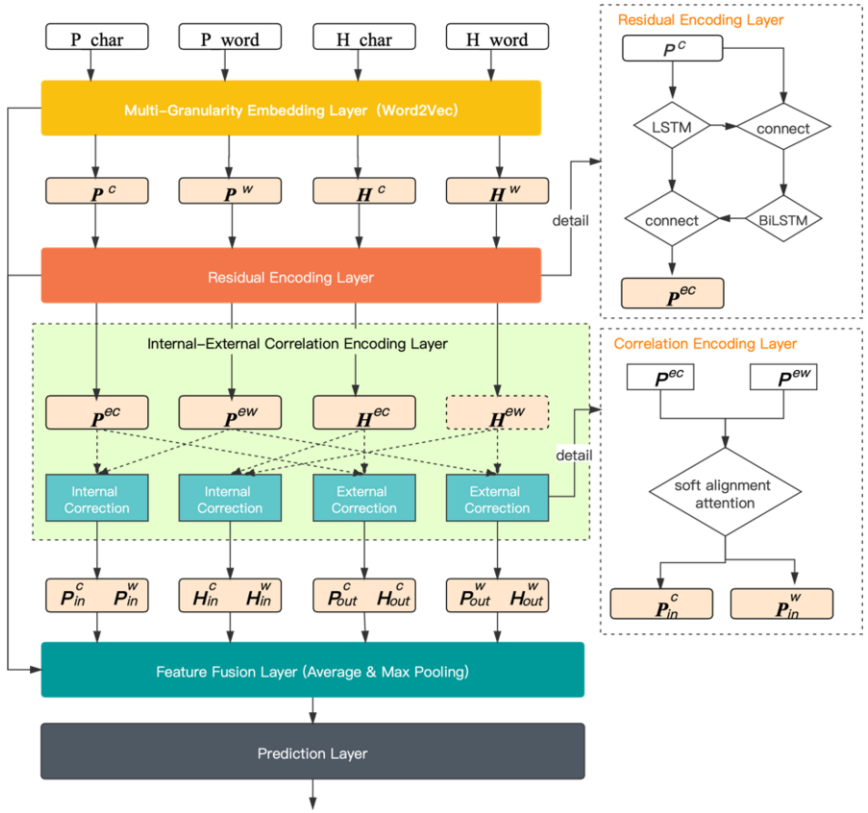

Figure 1. Architecture of the Multi-Granularity and Internal-External correlation Residual Model.

In this section, we introduce the proposed Multi-Granularity and Internal-External correlation Residual model (MGIER) for Chinese SSM.

Given two sentences P and H, we respectively segment P and H to character sequences P_char and H_char, and word sequences P_word and H_word. Then, the MGIER model inputs P_char, P_word, H_char, and H_word, and predicts a label $\hat{y}$ that indicates the semantic relationship between P and H. As shown in Fig. 1, the MGIER model architecture consists of five components: 1) a multi-granularity embedding layer, 2) a residual encoding layer, 3) an internal-external correlation encoding layer, 4) a global feature fusion layer, and 5) a prediction layer. We will detail each component in the following subsections.

3.1. Multi-Granularity Embedding Layer

In this layer, the character/word sequences P_char, P_word, H_char, and H_word of sentences P and H are first padded to the same length N. Then, an l-dimensional embedding vector for a character/word in the sequences is obtained by the pre-trained Word2Vec [20] on the data set, such as BQ [21] and LCQMC [22] in our experiments. It is noted that out-of-vocabulary (OOV) words are initialized with zero vectors. As a result, the sentences P and H are converted to four sequences P^c, P^w, H^c, H^w, where each sequence consists of N l-dimensional character/word embedding vectors. That is to say, each of P^c, P^w, H^c, H^w is a N*l matrix.

3.2. Residual Encoding Layer

In this layer, we use LSTM [23] and BiLSTM [24] to encode character/word embedding vector sequences in P^c, P^w, H^c, H^w obtained in the previous layer. The formula is shown in Eq. (1).

$$\left.\begin{aligned} P^{ec} &= [BiLSTM([LSTM(P^c), P^c]), LSTM(P^c)], \\ P^{ew} &= [BiLSTM([LSTM(P^w), P^w]), LSTM(P^w)], \\ H^{ec} &= [BiLSTM([LSTM(H^c), H^c]), LSTM(H^c)], \\ H^{ew} &= [BiLSTM([LSTM(H^w), H^w]), LSTM(H^w)]. \end{aligned}\right\} \quad (1)$$

Consider the first line in Eq. (1). First, we use LSTM to encode the character embedding vector sequence P^c, i.e., $\mathrm{LSTM}(P^c)$. Second, we combine $\mathrm{LSTM}(P^c)$ and P^c, i.e., $[\mathrm{LSTM}(P^c), P^c]$. Third, we use BiLSTM to encode the result in the second step, i.e., $\mathrm{BiLSTM}([\mathrm{LSTM}(P^c), P^c])$. Finally, we combine $\mathrm{BiLSTM}([\mathrm{LSTM}(P^c), P^c])$ and $\mathrm{LSTM}(P^c)$ and obtain P^{ec}. As shown in Eq. (1), similar steps are applied to the other character/embedding vector sequences P^w, H^c, H^w. The steps in Eq. (1) are detailed in the top-right of Fig. 1.

3.3. Internal and External Correlation Encoding Layer

In this layer, we employ soft alignment attention [1] to compute the internal correlation between characters/words in a sentence, and the external correlation between sentences.

Internal Correlation Encoding. We capture the internal correlation features between a character and a word in the same sentence.

We first compute the correlation weight via Eq. (2).

$$\left.\begin{aligned} p_{ij} &= (p_i^{ec})^{\mathrm{T}} p_j^{ew}, \quad i,j \in \{1,\dots,N\}, \\ h_{ij} &= (h_i^{ec})^{\mathrm{T}} h_j^{ew}, \quad i,j \in \{1,\dots,N\}. \end{aligned}\right\} \tag{2}$$

where p_i^{ec} and p_j^{ew} are respectively the i-th character encoding vector in P^{ec} and the j-th word encoding vector in P^{ew} for the sentence P, which are obtained in the previous layer. p_{ij} is the correlation weight between p_i^{ec} and p_j^{ew}. Similarly, h_i^{ec}, h_j^{ew}, and h_{ij} for the sentence H. After obtaining the correlation weights p_{ij} and h_{ij}, we compute the internal correlation features via Eq. (3).

$$\left.\begin{aligned} \overline{p_i^c} &= \sum_{j=1}^{N} \frac{\exp(p_{ij})}{\sum_{k=1}^{N} \exp(p_{ik})} p_j^{ew}, \quad i \in \{1,\dots,\mathrm{N}\}, \\ \overline{p_j^w} &= \sum_{j=1}^{N} \frac{\exp(p_{ij})}{\sum_{k=1}^{N} \exp(p_{ik})} p_i^{ec}, \quad j \in \{1,\dots,\mathrm{N}\}, \\ \overline{h_i^c} &= \sum_{j=1}^{N} \frac{\exp(h_{ij})}{\sum_{k=1}^{N} \exp(p_{ik})} h_j^{ew}, \quad i \in \{1,\dots,\mathrm{N}\}, \\ \overline{h_j^w} &= \sum_{j=1}^{N} \frac{\exp(h_{ij})}{\sum_{k=1}^{N} \exp(p_{ik})} h_i^{ec}, \quad j \in \{1,\dots,\mathrm{N}\}. \end{aligned}\right\} \tag{3}$$

where $\overline{p_i^c}$, $\overline{p_j^w}$, $\overline{h_i^c}$ and $\overline{h_j^w}$ are respectively the correlation features of the i-th character or the j-th word in the sentences P and H. With the above operations, we obtain the internal correlation features: P_{in}^c ($\left\{\overline{p_i^c}\right\}_{i=1}^{N}$), P_{in}^w ($\left\{\overline{p_j^w}\right\}_{j=1}^{N}$), H_{in}^c ($\left\{\overline{h_i^c}\right\}_{i=1}^{N}$), and H_{in}^w ($\left\{\overline{p_j^c}\right\}_{j=1}^{N}$).

External Correlation Encoding. We capture the external correlation features between two characters or two words in two different sentences.

We first compute the correlation weight via Eq. (4).

$$\left.\begin{aligned} c_{ii} &= (p_i^{ec})^{T} h_i^{ec}, \quad i \in \{1,\dots,\mathrm{N}\}, \\ w_{jj} &= (p_j^{ew})^{T} h_j^{ew}, \quad j \in \{1,\dots,\mathrm{N}\}. \end{aligned}\right\} \tag{4}$$

where p_i^{ec} and h_i^{ec} are respectively the i-th character encoding vector in P^{ec} for the sentence P and the i-th character encoding vector in H^{ec} for the sentence H. c_{ii} is the correlation weight between p_i^{ec} and h_i^{ec}. Similarly, p_j^{ew}, h_j^{ew} and w_{jj} for the word encoding vectors. After obtaining the correlation weights c_{ii} and w_{jj}, we compute the external correlation features via Eq. (5).

$$\left.\begin{aligned} \overline{p_i^c} &= \sum_{i=1}^{N} \frac{\exp(c_{ii})}{\sum_{k=1}^{N} \exp(c_{kk})} h_i^{ec}, \quad i \in \{1,\dots,\mathrm{N}\}, \\ \overline{p_j^w} &= \sum_{i=1}^{N} \frac{\exp(w_{jj})}{\sum_{k=1}^{N} \exp(\mathrm{w}_{kk})} h_j^{ew}, \quad j \in \{1,\dots,\mathrm{N}\}, \\ \overline{h_i^c} &= \sum_{i=1}^{N} \frac{\exp(c_{ii})}{\sum_{k=1}^{N} \exp(c_{kk})} p_i^{ec}, \quad i \in \{1,\dots,\mathrm{N}\}, \\ \overline{h_j^w} &= \sum_{i=1}^{N} \frac{\exp(w_{jj})}{\sum_{k=1}^{N} \exp(w_{kk})} p_j^{ew}, \quad j \in \{1,\dots,\mathrm{N}\}. \end{aligned}\right\} \tag{5}$$

where $\overline{p_i^c}$, $\overline{p_j^w}$, $\overline{h_i^c}$, and $\overline{h_j^w}$ are respectively the correlation features of characters or words in the sentences P and H. With the above operations, we obtain the external

correlation features: P_{out}^{c} ($\left\{\overline{p_i^c}\right\}_{i=1}^{N}$), P_{out}^{w} ($\left\{\overline{p_j^w}\right\}_{j=1}^{N}$), H_{out}^{c} ($\left\{\overline{h_i^c}\right\}_{i=1}^{N}$), and H_{out}^{w} ($\left\{\overline{p_j^c}\right\}_{j=1}^{N}$) .

3.4. Feature Fusion Layer

We have obtained the residual encoding features and internal-external correlation features in the above layers. Specifically, the residual encoding features are P^{ec}, P^{ew}, H^{ec}, and H^{ew}, the internal correlation features are P_{in}^{c} , P_{in}^{w} , H_{in}^{c} , and H_{in}^{w} , and the external correlation features are P_{out}^{c} , P_{out}^{w} , H_{out}^{c} , and H_{out}^{w}.

Internal Correlation Feature Fusion. The average and max pooling operations are able to extract a set of global and key features, respectively.

$$\left.\begin{aligned} In_P &= [avgpool([P^{\mathrm{ec}}; P^{ew}; P_{in}^{c}; P_{in}^{w}]);\ maxpool([P^{\mathrm{ec}}; P^{ew}; P_{in}^{c}; P_{in}^{w}])], \\ In_H &= [avgpool([H^{ec}; H^{ew}; H_{in}^{c}; H_{in}^{w}]);\ maxpool([H^{ec}; H^{ew}; H_{in}^{c}; H_{in}^{w}])], \\ In_{global} &= [avgPool([P_{in}^{c}; P_{in}^{w}; H_{in}^{c}; H_{in}^{w}]);\ maxpool([P_{in}^{c}; P_{in}^{w}; H_{in}^{c}; H_{in}^{w}])]. \end{aligned}\right\} \quad (6)$$

Consider the first line in Eq. (6). First, we combine the residual encoding features and internal correlation features of sentence P, i.e., $[P^{\mathrm{ec}}; P^{ew}; P_{in}^{c}; P_{in}^{w}]$. Second, we perform the average and max pooling operations on the combined feature. Finally, we combine the results of the pooling operations and obtain In_P which is the feature of P. As shown in Eq. (6), similar steps are applied to sentence H and obtain In_H which is the feature of H. For the global internal correlation feature In_{global}, we first combine all the internal correlation features, i.e., $[P_{in}^{c}; P_{in}^{w}; H_{in}^{c}; H_{in}^{w}]$, and then perform the pooling operations on the combined feature.

$$Sim_{in} = [|In_P - In_H|;\ In_P \times In_H]. \quad (7)$$

Here, the operations - and × are performed element-wise to infer the relationship between two sentences. As shown in Eq. (7), we compute the correlation feature Sim_{in} for the tuple <In_P, In_H>.

$$In = [Sim_{in};\ In_{global}]. \quad (8)$$

As shown in Eq. (8), we obtain the internal correlation feature In by combining the correlation feature Sim_{in} and the global internal correlation feature In_{global}.

External Correlation Features Fusion. We integrate the residual encoding features and the external correlation features.

$$\left.\begin{aligned} Out_w &= [avgpool([P^{\mathrm{ew}}; H^{ew}; P_{out}^{w}; H_{out}^{w}]);\ maxpool([P^{\mathrm{ew}}; H^{ew}; P_{out}^{w}; H_{out}^{w}])], \\ Out_c &= [avgpool([P^{ec}; H^{ec}; P_{out}^{c}; H_{out}^{c}]);\ maxpool([H^{ec}; H^{ew}; P_{out}^{c}; H_{out}^{c}])], \\ Out_{global} &= [avgPool([P_{out}^{c}; P_{out}^{w}; H_{out}^{c}; H_{out}^{w}]);\ maxpool([P_{out}^{c}; P_{out}^{w}; H_{out}^{c}; H_{out}^{w}])]. \end{aligned}\right\} \quad (9)$$

Consider the first line in Eq. (9). First, we combine the residual encoding features and external correlation features of word-granularity for sentences P and H, i.e., $[P^{\mathrm{ew}}; H^{ew}; P^{w}_{out}; H^{w}_{out}]$. Second, we perform the average and max pooling operations on the combined feature. Finally, we combine the results of the pooling operations and obtain Out_w, which is the word-granularity feature for sentences P and H. Similar steps are applied to character-granularity for sentences P and H. For the global external correlation feature Out_{global}, we first combine all the external correlation features, i.e., $[P^{c}_{out}; P^{w}_{out}; H^{c}_{out}; H^{w}_{out}]$, and then perform pooling operations on the combined feature.

$$Sim_{out} = [|Out_w - Out_c|; Out_w \times Out_c\ ]. \tag{10}$$

As shown in Eq. (10), we compute the correlation feature Sim_{out} for the tuple <Out_{w}, Out_c >.

$$Out\ =\ [Sim_{out};\ Out_{global}]. \tag{11}$$

As shown in Eq. (11), we obtain the external correlation feature Out by combining the correlation feature Sim_{out} and the global external correlation feature Out_{global}.

$$Feature_{total}\ =\ [In; Out]. \tag{12}$$

Finally, we obtain the global feature $Feature_{total}$ by combining the internal/external correlation features, as shown in Eq. (12), which is the input of the prediction layer.

3.5. Prediction Layer

The prediction layer is a multi-layer perceptron classifier with three dense sub-layers, in which the first two dense layers are activated with the ReLU function [25] and the last dense layer is connected with the sigmoid activation function in our experiments.

3.6. Loss Function

When training the models, we use two loss functions which have different effects on different datasets. The first one is the modified binary cross-entropy l_{modify} [26] shown in Eq. (13), which can focus on the training samples that are not easy to distinguish.

$$l_{modify} = -\sum \lambda(y_{true}, y_{pred})(y_{true} log y_{pred} + (1 - y_{true}) log(1 - y_{pred})). \tag{13}$$

where y_{true} and y_{pred} are the true value and the predicted value, $\lambda(y_{true,}y_{pred})$ is defined as in Eq. (14).

$$\lambda(y_{true}, y_{pred}) = 1 - \theta(y_{true} - m)\theta(y_{pred} - m) - \theta(1 - m - y_{true})\theta(1 - m - y_{pred}). \tag{14}$$

where m is a threshold which is usually set to 0.7, $\theta(x)$ is the unit step function and is defined as shown in Eq. (15).

$$\theta(x) = \begin{cases} 1, & x > 0, \\ 1/2, & x = 0, \\ 0, & x < 0. \end{cases} \tag{15}$$

The second loss function is the equilibrium binary cross-entropy $l_{equilibrium}$ [7] shown in Eq. (16), which can strengthen its ability to distinguish the fuzzy boundary and eliminate the blurring phenomenon in classification tasks.

$$l_{equilibrium} = -\sum_{i=1}^{n}(l_{mse} \times y_{true} log y_{pred} + (1 - l_{mse}) \times (1 - y_{true}) log(1 - y_{pred})). \tag{16}$$

Where n is the count of samples and l_{mse} is the equilibrium factor MSE, defined as in Eq. (17).

$$l_{mse} = \frac{1}{2n}\sum_{i=1}^{n}(y_{true} - y_{pred})^2. \tag{17}$$

4. Experiments

The goal of the experiments is to evaluate the effect of the performance of the proposed model, MGIER, for Chinese SSM.

4.1. Experiment Setup

Datasets. In the experiments, we use two Chinese datasets BQ [21] and LCQMC [22]. BQ is a Chinese bank question corpus for sentence intention equivalence identification, which comes from Weizhong bank's online customer service logs. LCQMC is a Chinese question matching corpus collected from Baidu Knows. The two datasets consist of a large number of instances in the form of (P, H, Label), where P and H are two Chinese sentences, and Label indicates whether P and H have the same meaning. We split each dataset into a training set, validation set, and test set by the same proportion mentioned in [24, 25]. A summary of the datasets is shown in Table 2.

Table 2. Experimental data sets

Dataset	Source	Scale(train/validation/test)	Positive:Negtive
BQ	Weizhong bank	100,000/10,000/10,000	1:1
LCQMC	Baidu Knows	238,766/8802/12,499	1.35:1

Baseline models. In the experiments, we compare the proposed model with seven baseline models. ESIM [1] employs soft alignment attention to capture deep correlation information between two sentences. Although MGF [7] and ICE [8] use multi-granularity to enhance text representation, only ICE uses soft alignment attention to capture the correlation information of sentences. Both have good performance for Chinese SSM. DRCN [9] utilizes the representational power of recurrent networks and attentive information. ESIM and DRCN also have remarkable performance in English SSM. BiMPM [14] matches two sentences from different directions. ARC-II [27] is a simplification of MatchPyramid. MatchPyramid [28] comes from modeling text matching as image recognition, which takes the matching matrix as an image.

Metrics. The metrics used in the experiments include accuracy, precision, recall and F1_score.

Parameters. In the experiments, the hidden states of LSTM and BILSTM and the character/word embedding vectors have 300 dimensions. For LSTM, the dropout rates 0.4 and 0 are used for the datasets BQ and LCQMC, respectively. In BiLSTM, the dropout rates are set to 0.55 and 0.25, respectively, for BQ and LCQMC. The dropout rates 0.3 and 0.5 are used for BQ and LCQMC, respectively, in the prediction layer which consists of two densely-connected hidden layers with 600 units and one classifier with a sigmoid activation function. When training the models, we use Adam optimizer, where the number of epochs is 100 and the batch size is 512.

Environment Setting. We implemented the proposed model using Python with the Keras and Tensorflow frameworks, and the baseline models using the Pytorch frameworks. All experiments are run on a workstation equipped with two Intel (R) Xeon (R) gold 6132 CPU, @ 2.60GHz 256GB memory, four pieces of NVIDIA Tesla V100 SXM2 32GB GPU, and CentOS Linux.

4.2. Experimental Results

The effect of loss functions. We compare the effect of two loss functions: the modified binary cross-entropy l_{modify} defined in Eq. (14) and the equilibrium cross-entropy $l_{equilibrium}$ defined in Eq. (18). The results are shown Table 3. From this table, we can see that the modified binary cross-entropy l_{modify} is better than the equilibrium cross-entropy $l_{equilibrium}$ for the dataset BQ, while $l_{equilibrium}$ is better than l_{modify} for the dataset LCQMC. The reasons may be that the modified binary cross-entropy focuses on the training samples that are not easy to distinguish, and the equilibrium cross-entropy distinguishes the fuzzy boundary and eliminates the blurring phenomenon. As a result, we use the modified binary cross-entropy as the loss function for BQ, and the equilibrium cross-entropy for LCQMC.

Table 3. The effect of loss functions

Dataset	Loss Function	Accuracy	Precision	Recall	F1_score
BQ	l_{modify} [8]	**84.93**	**84.96**	**84.94**	**84.87**
	$l_{equilibrium}$[7]	81.60	84.97	76.78	80.56
LCQMC	l_{modify} [8]	85.73	80.64	94.04	86.78
	$l_{equilibrium}$ [7]	**87.70**	**88.77**	86.49	**87.57**

The effect of models. We compare the proposed model with the baseline models. The results are shown in Table 4. In most cases, the accuracy, precision, and F1_score of the models using character/word granularity are better than that of other models. This indicates that using multi-granularity can improve the performance of the model. Among the models, the performance of MGIER, ESIM, and ICE using soft alignment attention is also better than that of other models in most cases. Thus, soft alignment attention is beneficial to Chinese SSM. Generally, the accuracy of MGIER, outperforms that of the baseline models for the dataset LCQMC and BQ. This is because MGIER uses multi-granularity and soft alignment attention to capture the fine-

grained information and internal/external correlation information. Also, MGIER employs residual connections to preserve semantic information.

Table 4. The performance of models

Dataset	Model	Accuracy	Precision	Recall	F1_score
BQ	$MGIER_{char+word}$	**84.93**	84.96	84.94	84.87
	$ESIM_{char}$ [1]	83.69	78.19	94.22	85.46
	$ESIM_{word}$	81.59	83.87	78.74	81.23
	$MGF_{char+word}$ [7]	82.61	89.65	73.75	80.83
	$ICE_{char+word}$ [8]	84.05	83.31	85.10	84.12
	$DRCN_{char+word}$ [9]	76.78	78.13	74.38	76.21
	$BiMPM_{char}$ [14]	82.23	80.41	85.74	82.99
	$ARC\text{-}II_{char}$[27]	76.68	75.94	78.10	77.01
	$MatchPyramid_{char}$[28]	67.09	65.47	72.35	68.74
LCQMC	$MGIER_{char+word}$	**87.70**	**88.77**	86.49	**87.57**
	$ESIM_{char}$	84.83	78.19	94.22	85.46
	$ESIM_{word}$	83.69	75.50	94.42	83.90
	$MGF_{char+word}$	85.86	81.45	92.89	86.76
	$ICE_{char+word}$	86.15	81.93	92.70	86.94
	$DRCN_{char+word}$	78.93	72.10	94.42	81.76
	$BiMPM_{char}$	82.23	75.53	95.47	84.34
	$ARC\text{-}II_{char}$	82.35	88.53	74.34	80.81
	$MatchPyramid_{char}$	72.47	67.01	88.43	76.26

Comparison with pre-trained models. We compare MGIER with the pre-trained models as shown in Table 5. From the results in Table 5, we can see that the accuracy of MGIER is higher than that of other pre-trained models, with fewer parameters. This is probably because MGIER can capture more and deeper semantic information.

Table 5. Comparison with pre-trained models

Dataset	Model	Accuracy/Parameters	Dataset	Model	Accuracy/Parameters
BQ	MGIER	**84.93 (21.25M)**	LCQMC	MGIER	**87.70 (21.25M)**
	Bert	83.23 (169.62M)		Bert	87.57 (169.62M)
	NEZHA-Base	84.79 (97.16M)		NEZHA-Base	86.07 (97.16M)
	NEZHA-Base-WWM	84.67 (97.16M)		NEZHA-Base-WWM	86.35 (97.16M)
	Roberta-wwm-ext	84.11 (97.16M)		Roberta-wwm-ext	84.86 (97.16M)

Ablation Experiments. We conduct an ablation study for the proposed model MGIER on the datasets BQ and LCQMC. The results in Table 6 demonstrate the effectiveness of the residual encoding layer as well as the internal-external correlation encoding layer.

First, we remove the residual encoding and only preserve the output features of BiLSTM and LSTM, denoted case (1). Table 6 shows that the accuracy, precision, and F1_score drops obviously, which demonstrates the effectiveness of residual encoding.

Next, we remove the internal correlation encoding from MGIER, denoted case (2). The precision drops to 82.77% and 84.65% for the datasets BQ and LCQMC, respectively. We remove the external correlation encoding from MGIER, denoted case (3), and the recall drops to 79.36% for BQ, and the precision drops to 87.3% for LCQMC. This suggests that computing the internal/external correlation of the sentences is useful to improve the performance of the model.

Table 6. Ablation experiments

Dataset	Model	Accuracy	Precision	Recall	F1_score
BQ	MGIER	**84.93**	84.96	**84.94**	**84.87**
	(1) No residual encoding	83.76	84.51	82.62	83.47
	(2) No internal encoding	83.48	82.77	84.55	83.57
	(3) No external encoding	82.81	85.19	79.36	82.11
LCQMC	MGIER	**87.70**	**88.77**	86.49	87.57
	(1) No residual encoding	84.35	78.86	93.84	85.65
	(2) No internal encoding	86.81	84.65	89.88	87.15
	(3) No external encoding	87.65	87.30	88.14	87.67

5. Conclusions

We propose a Chinese SSM model, MGIER, to capture deep semantic information from the bottom embedding layer to the top prediction layer. Experimental results for two Chinese datasets demonstrate that the proposed method outperforms the state-of-the-art methods. Moreover, the proposed method is more efficient than pre-trained models. Although the MGIER model has achieved remarkable performance, there are still some important points that need to be considered. Through analyzing the incorrectly-predicted examples by the MGIER model, we find that different parts of a sentence have greater influence on the semantic matching results. For example, if the subject of the sentence does not match but the rest of the sentence does match, the result predicted by the MGIER model is matching when in fact it does not. Moreover, the MGIER model just considers the character and word granularity, but the phrases and themes of the sentence also have important semantic information. In the future, we will attempt to utilize phrases, themes, and subject-verb-objects of sentences to further improve the performance of MGIER for Chinese SSM.

References

[1] CHEN Q, ZHU X, LING Z, et al.: Enhanced LSTM for natural language inference[C]. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, 1657-166.

[2] YANG R, J ZHANG, GAO X, et al.: Simple and Effective Text Matching with Richer Alignment Features[J]. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019.

[3] MUELLER J, THYAGARAJAN A.: Siamese recurrent architectures for learning sentence similarity[C]. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, 2786-2792.
[4] Kun Zhang, Guangyi Lv, Linyuan Wang, Le Wu, Enhong Chen, Fangzhao Wu, and Xing Xie. 2019. DRr-Net: Dynamic Re-read Network for Sentence Semantic Matching. (2019).
[5] Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural language inference over interaction space. In Proceedings of the International Conference on Learning Representations.
[6] Jiangping Huang, Shuxin Yao, Chen Lyu, and Donghong Ji. 2017. Multi-granularity neural sentence model for measuring short text similarity. In Proceedings of the International Conference on Database Systems for Advanced Applications, pages 439–455.
[7] ZHANG X, LU W, ZHANG G, et al.: Chinese sentence semantic matching based on multi-granularity fusion model[M]. 2020.
[8] ZHANG X, LI Y, LU W, et al.: Intra-Correlation Encoding for Chinese Sentence Intention Matching[C]// Proceedings of the 28th International Conference on Computational Linguistics. 2020.
[9] Kim S, Kang I, Kwak N. Semantic Sentence Matching with Densely-connected Recurrent and Co-attentive Information[J]. 2018.
[10] HUANG P S, HE X D, GAO J F, et al.: Learning deep structured semantic models for web search using clickthrough data［C］//Proceedings of the 22nd ACM International Conference on Conference on Information＆Knowledge Management, 2013: 2333-2338.
[11] SHEN Y, HE X, GAO J, et al.: A latent semantic model with convolutional-pooling structure for information retrieval[C]. Proceedings of the 23rd ACM international conference on conference on information and knowledge management, 2014, 101-110.
[12] PALANGI H, DENG L, SHEN Y, et al.: Semantic modelling with long-short-term memory for information retrieval[J]. arXiv preprint arXiv:1412.6629, 2014.
[13] PENG S, CUI H, XIE N, et al. Enhanced-RCNN: An Efficient Method for Learning Sentence Similarity[C]// WWW '20: The Web Conference 2020. 2020.
[14] WANG Z, HAMZA W, FLORIAN R.: Bilateral Multi-Perspective Matching for Natural Language Sentences[C]// Twenty-Sixth International Joint Conference on Artificial Intelligence. 2017.
[15] DEVLIN J, CHANG M, LEE K, et al.: BERT: pre-training of deep bidirectional transformers for language understanding[C]. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, 2019, 4171-4186
[16] LIU W, ZHOU P, ZHAO Z, et al. K-bert:enabling language representation with knowledge graph[J]. arXiv preprint arXiv:1909.07606, 2019
[17] WEI J, REN X, LI X, et al. NEZHA: Neural Contextualized Representation for Chinese Language Understanding[J]. 2019.
[18] REIMERS N, GUREVYCH I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks[J]. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.
[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.
[20] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C]. Advances in neural information processing systems, 2013, 3111-3119
[21] CHEN J, CHEN Q C, LIU X, et al. 2018. The BQ corpus: A large-scale domain-specific Chinese corpus for sentence semantic equivalence identification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 4946–4951.
[22] LIU X, CHEN Q C, DENG C, et al. 2018. LCQMC: A large-scale Chinese question matching corpus. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1952–1962.
[23] Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 551–561.
[24] Schuster, Mike, Paliwal, et al. Bidirectional recurrent neural networks. [J]. IEEE Transactions on Signal Processing, 1997.
[25] Nair V, Hinton G E. Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair[C]// International Conference on International Conference on Machine Learning. Omnipress, 2010.
[26] SU J L. 2017. Text emotion classification (IV): Better loss function. Web page. https://spaces.ac.cn/archives/4293.
[27] Hu, B.; Lu, Z.; Li, H.; and Chen, Q. 2014. Convolutional neural network architectures for matching natural language sentences. In Advances in Neural Information Processing Systems, 2042–2050.
[28] Liang P, Lan Y, Guo J, et al. Text Matching as Image Recognition. 2016.