Intra-Correlation Encoding for Chinese Sentence Intention Matching

Xu Zhang[1], Yifeng Li[2], Wenpeng Lu[1]*, Ping Jian[3], Guoqiang Zhang[4]
[1]School of Computer Science and Technology,
Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
[2]Department of Computer Science,
Brock University, Niagara Region, Canada
[3]School of Computer Science and Technology,
Beijing Institute of Technology, Beijing, China
[4]Centre for Audio, Acoustics and Vibration,
University of Technology Sydney, Sydney, Australia
xuzhang.p@foxmail.com, yli2@brocku.ca, lwp@qlu.edu.cn,
pjian@bit.edu.cn, guoqiang.zhang@uts.edu.au

Abstract

Sentence intention matching is vital for natural language understanding. Especially for Chinese sentence intention matching task, due to the ambiguity of Chinese words, semantic missing or semantic confusion are more likely to occur in the encoding process. Although the existing methods have enriched text representation through pre-trained word embedding to solve this problem, due to the particularity of Chinese text, different granularities of pre-trained word embedding will affect the semantic description of a piece of text. In this paper, we propose an effective approach that combines character-granularity and word-granularity features to perform sentence intention matching, and we utilize soft alignment attention to enhance the local information of sentences on the corresponding levels. The proposed method can capture sentence feature information from multiple perspectives and correlation information between different levels of sentences. By evaluating on BQ and LCQMC datasets, our model has achieved remarkable results, and demonstrates better or comparable performance with BERT-based models.

1 Introduction

As a branch of sentence semantic matching (SSM), sentence intention matching (SIM) is critical to question answering systems in certain applications. In general, SSM is to judge whether two sentences express the same meaning. However, in a question answering system, SIM intends to determine whether two questions share the same intention and could be addressed by the same answer, which is more challenging than other SSM tasks. As an example shown in Table 1, although both sentences in $Q1$ and $Q2$ share similar intention in fact, it is difficult to distinguish whether they are similar at the semantic intention level without considering the deep context.

With the development of deep learning, a series of SSM models are proposed for semantic matching tasks (Wang et al., 2017; Gong et al., 2018; Huang et al., 2019; Li et al., 2019; Liu et al., 2020). However, these models simply consider the characteristics of the semantic level of the text but overlook the deep intentional features. Researchers have attempted to extract deeper semantic features through attention mechanisms (Tan et al., 2018; Tay et al., 2018), memory networks (Cheng et al., 2016), as well as the addition of external syntactic structures and lexical datasets as in WordNet (Chen et al., 2017). Although the above methods obtain deep semantic features from different perspectives, they cannot completely overcome the feature missing phenomenon in the encoding process. Especially due to the diversity of Chinese semantic features, the above existing methods cannot better capture complicated deep semantic features.

*Corresponding Author: Wenpeng Lu

Proceedings of the 28th International Conference on Computational Linguistics, pages 5193–5204
Barcelona, Spain (Online), December 8-13, 2020

Table 1: Examples from BQ corpus.

Sentence Pairs
*Q*1: 我提交申请时，说身份信息输入错误次数过多 EN: When I submitted my application, I was notified that my identity information was inputted incorrectly too many times
*Q*2: 我的微粒贷怎么申请时显示身份信息输入错误 EN: Why did it show that my identity information was inputted incorrectly when I applied for my Webank loan

In English SSM tasks, Wang et al. and Gong et al. employ multi-granularity fusion to extract the corresponding richer semantic features, where the more fine-grained character embeddings are employed together with the traditional word embeddings (Wang et al., 2017; Gong et al., 2018). Although the introduction of character embeddings is beneficial to enrich English text representation, one single English character is hard to express a special meaning. Different from English, a Chinese character is able to represent a solid meaning, which can convey more semantic features and information. Thus, there should be great interest and potential to explore and combine multi-granularity embeddings for Chinese SSM tasks. Huang et al. (Huang et al., 2017) and Zhang et al. (Zhang et al., 2020) achieve better performance by combining character, word, and other granularities to obtain semantic encoding features from Chinese text. The multi-granularity fusion method can be applied to extract the semantic features of a text sequence, which can effectively alleviate the phenomenon of missing semantic features in the encoding process.

In this paper, inspired by the existing work, we push forward this line of research by proposing a better multi-granularity fusion approach to capture semantic features from text sequences. In (Huang et al., 2017) and (Zhang et al., 2020), the encoding features in multi-perspective granularities are integrated to generate the final text encodings. However, the correlation and distinction between text features on different granularities are not considered, thus the corresponding semantic features are not further explored in these works. Sequential inference models based on chain LSTMs are implemented in (Chen et al., 2017), enabling the capture of more features from different perspectives. They incorporate syntactic parsing information in tree LSTM into the classic BiLSTM model with the help of soft alignment attention. In our work, in order to better extract the correlations between different granularities in SIM, inspired by the work in (Chen et al., 2017), we employ soft alignment attention to enhance local information representation between different granularities and capture more sentence correlation.

Our contributions are summarized as follows:

- We propose a novel sentence intention matching model, named **intra-correlation encoding model (ICE)**, to better extract sentence intention features. It can capture sentence feature information from multiple perspectives and the correlation information between sentences on character-granularity and word-granularity.

- We propose a novel deep neural architecture for sentence intention matching task, which includes a multi-granularity embedding layer, an intra-correlation encoding layer, a global inference composition layer, and a prediction layer. Our source code is publicly available on GitHub [1]. This work may provide a new reference for researchers in the NLP community.

The rest of the paper is structured as follows. We describe our novel sentence intention matching model in Section 2. Section 3 demonstrates the experimental results. Related work is introduced in Section 4, followed by conclusions in Section 5.

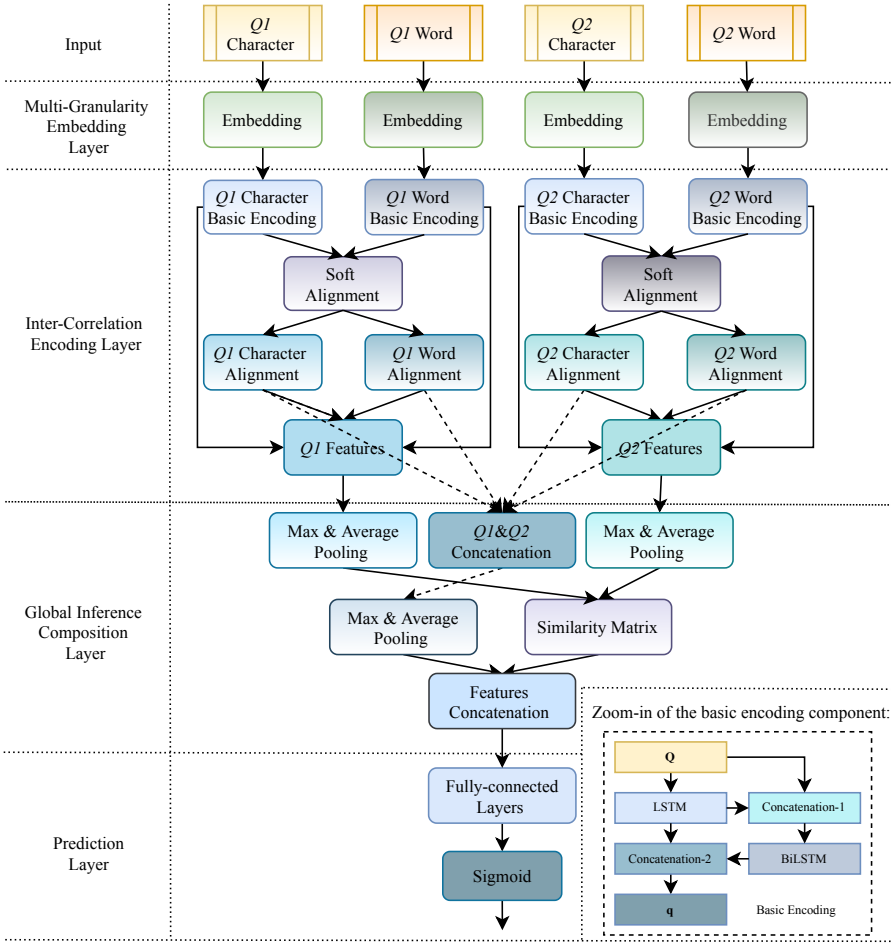

Figure 1: Architecture of our intra-correlation encoding model.

2 Model

We propose a multi-granularity (character-granularity and word-granularity) neural sentence model, whose architecture is shown in Figure 1. We utilize the siamese network structure in the SIM task. Our model architecture consists of four parts: a multi-granularity embedding layer, an intra-correlation encoding layer, a global inference composition layer, and a prediction layer. In the following subsections, we first describe the initial representation, including word-granularity and character-granularity embeddings, then introduce the intra-correlation encoding layer in detail. Next, we describe the global inference composition layer, which measures the feature information between two sentence representations. Finally, the prediction layer is introduced, which predicts whether the corresponding sentences match each other in semantic intention.

2.1 Multi-Granularity Embedding Layer

In our notation, we have two Chinese sentences $Q1$=($q1_1, q1_2, \cdots, q1_i$) and $Q2$=($q2_1, q2_2, \cdots, q2_j$). We employ different segmentation methods to segment $Q1$ for character and word [2], and obtain the multi-granularity sentence representation of character-based $Q1$ and word-based $Q1$. An example is shown in Table 2. Sentence $Q2$ is processed in the same way. The character-based and word-based sentences are padded to the same length N. Corresponding embedings at the character and word levels are obtained by pre-training Word2Vec (Mikolov et al., 2013) on the target dataset, such as BQ or LCQMC in our experiments.

[1]https://github.com/XuZhangp/ICE
[2]https://github.com/fxsjy/jieba

Table 2: Examples of multi-granularity based sentences.

Sentence Pairs
Sentence $Q1$: 我提交申请时，说身份信息输入错误次数过多
Character-based $Q1$: 我/提/交/申/请/时/，/说/身/份/信/息/输/入/错/误/次/数/过/多/
Word-based $Q1$: 我/提交/申请/时/，/说/身份/信息/输入/错误/次数/过多/
Sentence $Q2$: 我的微粒贷怎么申请时显示身份信息输入错误
Character-based $Q2$: 我/的/微/粒/贷/怎/么/申/请/时/显/示/身/份/信/息/输/入/错/误/
Word-based $Q2$: 我/的/微粒贷/怎么/申请/时/显示/身份/信息/输入/错误/

2.2 Intra-Correlation Encoding Layer

LSTM and BiLSTM are utilized to encode input sentences $Q1$ and $Q2$ at their character and word level granularities, respectively, as shown in Equations 1 and 2.

$$\begin{aligned}\mathbf{q1}_n^c &= [\mathrm{BiLSTM}([\mathrm{LSTM}(\mathbf{Q1}^{\mathrm{c}}), \mathbf{Q1}^{\mathrm{c}}], \mathrm{n}); \mathrm{LSTM}(\mathbf{Q1}^{\mathrm{c}}, \mathrm{n})], \qquad \mathrm{n} \in (1, 2, \cdots, \mathrm{N}), \\ \mathbf{q1}_m^w &= [\mathrm{BiLSTM}([\mathrm{LSTM}(\mathbf{Q1}^{\mathrm{w}}), \mathbf{Q1}^{\mathrm{w}}], \mathrm{m}); \mathrm{LSTM}(\mathbf{Q1}^{\mathrm{w}}, \mathrm{m})], \qquad \mathrm{m} \in (1, 2, \cdots, \mathrm{N}),\end{aligned} \tag{1}$$

where, we utilize $\mathbf{q1}_n^c$ and $\mathbf{q1}_m^w$ to represent the hidden (output) state generated by the basic encoding module for the n-th character and m-th word, respectively. The same is applied to $\mathbf{q2}_n^c$ and $\mathbf{q2}_m^w$. In this way, we generate multi-granularity representations of encoding features for two sentences with the basic encoding components.

As shown in Equation 1 for character-based $Q1$, the LSTM Layer is the first layer in the basic encoding components after the multi-granularity embedding layer. Next, the concatenated outputs from the multi-granularity embedding layer ($\mathbf{Q1}^c$) and the LSTM layer ($\mathrm{LSTM}(\mathbf{Q1}^c)$) flow to the BiLSTM layer. Finally, the outputs of the BiLSTM layer ($\mathrm{BiLSTM}([\mathrm{LSTM}(\mathbf{Q1}^{\mathrm{c}}), \mathbf{Q1}^{\mathrm{c}}])$) and the LSTM layer ($\mathrm{LSTM}(\mathbf{Q1}^c)$) are combined as the final feature representation. Our model follows a siamese network structure, which applies the same encoding method to word-based $Q1$, word-based $Q2$, and character-based $Q2$ as shown in Equations 1 and 2.

$$\begin{aligned}\mathbf{q2}_n^c &= [\mathrm{BiLSTM}([\mathrm{LSTM}(\mathbf{Q2}^{\mathrm{c}}), \mathbf{Q2}^{\mathrm{c}}], \mathrm{n}); \mathrm{LSTM}(\mathbf{Q2}^{\mathrm{c}}, \mathrm{n})], \qquad \mathrm{n} \in (1, 2, \cdots, \mathrm{N}), \\ \mathbf{q2}_m^w &= [\mathrm{BiLSTM}([\mathrm{LSTM}(\mathbf{Q2}^{\mathrm{w}}), \mathbf{Q2}^{\mathrm{w}}], \mathrm{m}); \mathrm{LSTM}(\mathbf{Q2}^{\mathrm{w}}, \mathrm{m})], \qquad \mathrm{m} \in (1, 2, \cdots, \mathrm{N}).\end{aligned} \tag{2}$$

In order to capture the correlation information between different granularities of the same sentence, we employ the soft alignment attention work of Chen et al. on the text semantic matching task (Chen et al., 2017). They utilize the attention mechanism to compute the attention weights as the similarity of a hidden state tuple between a premise and a hypothesis.

From this inspiration, we compute the attention weights $\mathbf{eq1}_{nm}$ as the similarity of a hidden state tuple $<\mathbf{q1}_n^c, \mathbf{q1}_m^w>$ for $Q1$ between character-granularity and word-granularity, as shown in Equation 3, where $\mathbf{q1}_n^c$ and $\mathbf{q1}_m^w$ are computed earlier in Equations 1 and 2. The same is applied to $Q2$, as shown in Equation 4. In this way, we can obtain the text feature correlation between different granularities and extract more abundant semantic features.

$$\mathbf{eq1}_{nm} = \mathbf{q1}_n^{cT} \mathbf{q1}_m^w, \qquad n, m \in (1, 2, \cdots, N), \tag{3}$$

$$\mathbf{eq2}_{nm} = \mathbf{q2}_n^{cT} \mathbf{q2}_m^w, \qquad n, m \in (1, 2, \cdots, N). \tag{4}$$

Through the above Equations 3 and 4, we obtain the correlation attention weights for sentence features on different granularities, i.e., $\mathbf{eq1}_{nm}$ and $\mathbf{eq2}_{nm}$. For the hidden state of the n-th character in character-based $Q1$, i.e., $\mathbf{q1}_n^c$, its correlation semantics in the word-based $Q1$ is identified based on $\mathbf{eq1}_{nm}$, as shown in Equation 5.

$$\overline{\mathbf{q1}}_n^c = \sum_{m=1}^{N} \frac{\exp(\mathbf{eq1}_{nm})}{\sum_{k=1}^{N} \exp(\mathbf{eq1}_{nk})} \mathbf{q1}_m^w, \qquad n \in (1, 2, \cdots, N), \tag{5}$$

where $\overline{\mathbf{q1}}_n^c$ is a weighted summation of $\{\mathbf{q1}_m^w\}_{m=1}^N$. Intuitively, the content in $\{\mathbf{q1}_m^w\}_{m=1}^N$ that is relevant to $\mathbf{q1}_n^c$ will be selected and represented as $\overline{\mathbf{q1}}_n^c$. The same is performed for each word represented in the word-based $Q1$ with Equation 6.

$$\overline{\mathbf{q1}}_m^w = \sum_{n=1}^{N} \frac{\exp(\mathbf{eq1}_{nm})}{\sum_{k=1}^{N} \exp(\mathbf{eq1}_{km})} \mathbf{q1}_n^c, \qquad m \in (1, 2, \cdots, N). \tag{6}$$

Using Equations 3 - 6, we obtain the correlation feature expressions $\overline{\mathbf{q1}}_n^c$ and $\overline{\mathbf{q1}}_m^w$ for sentence $Q1$. Similarly, we can obtain the correlation features for sentence $Q2$, using:

$$\begin{aligned} \overline{\mathbf{q2}}_n^c &= \sum_{m=1}^{N} \frac{\exp(\mathbf{eq2}_{nm})}{\sum_{k=1}^{N} \exp(\mathbf{eq2}_{nk})} \mathbf{q2}_m^w, \qquad n \in (1, 2, \cdots, N), \\ \overline{\mathbf{q2}}_m^w &= \sum_{n=1}^{N} \frac{\exp(\mathbf{eq2}_{nm})}{\sum_{k=1}^{N} \exp(\mathbf{eq2}_{km})} \mathbf{q2}_n^c, \qquad m \in (1, 2, \cdots, N). \end{aligned} \tag{7}$$

With the above operations, we generate the sentence feature representations $\mathbf{q1}^c$ ($\{\mathbf{q1}_n^c\}_{n=1}^N$), $\mathbf{q1}^w$ ($\{\mathbf{q1}_m^w\}_{m=1}^N$), $\mathbf{q2}^c$ ($\{\mathbf{q2}_n^c\}_{n=1}^N$), and $\mathbf{q2}^w$ ($\{\mathbf{q2}_m^w\}_{m=1}^N$) for sentences $Q1$ and $Q2$. In addition, we also generate the feature representations of correlations $\overline{\mathbf{q1}}^c$ ($\{\overline{\mathbf{q1}}_n^c\}_{n=1}^N$), $\overline{\mathbf{q1}}^w$ ($\{\overline{\mathbf{q1}}_m^w\}_{m=1}^N$), $\overline{\mathbf{q2}}^c$ ($\{\overline{\mathbf{q2}}_n^c\}_{n=1}^N$), and $\overline{\mathbf{q2}}^w$ ($\{\overline{\mathbf{q2}}_m^w\}_{m=1}^N$) between multi-granularity sentences.

2.3 Global Inference Composition Layer

Thus far we have obtained a series of basic and correlation encoding feature representations through the intra-correlation encoding layer. We now apply average and max pooling operations on them and obtain the final feature representations for $Q1$ ($\mathbf{q1}_{avg}^c$, $\mathbf{q1}_{avg}^w$, $\overline{\mathbf{q1}}_{avg}^c$, $\overline{\mathbf{q1}}_{avg}^w$, $\mathbf{q1}_{max}^c$, $\mathbf{q1}_{max}^w$, $\overline{\mathbf{q1}}_{max}^c$, $\overline{\mathbf{q1}}_{max}^w$) and $Q2$ ($\mathbf{q2}_{avg}^c$, $\mathbf{q2}_{avg}^w$, $\overline{\mathbf{q2}}_{avg}^c$, $\overline{\mathbf{q2}}_{avg}^w$, $\mathbf{q2}_{max}^c$, $\mathbf{q2}_{max}^w$, $\overline{\mathbf{q2}}_{max}^c$, $\overline{\mathbf{q2}}_{max}^w$) as shown in Equations 8 - 11.

$$\mathbf{q1}_{avg}^c = \frac{1}{N}\sum_{n=1}^{N} \mathbf{q1}_n^c, \quad \mathbf{q1}_{max}^c = \max_{\text{along axis } n} \mathbf{q1}^c, \quad \mathbf{q1}_{avg}^w = \frac{1}{N}\sum_{m=1}^{N} \mathbf{q1}_m^w, \quad \mathbf{q1}_{max}^w = \max_{\text{along axis } m} \mathbf{q1}^w, \tag{8}$$

$$\overline{\mathbf{q1}}_{avg}^c = \frac{1}{N}\sum_{n=1}^{N} \overline{\mathbf{q1}}_n^c, \quad \overline{\mathbf{q1}}_{max}^c = \max_{\text{along axis } n} \overline{\mathbf{q1}}^c, \quad \overline{\mathbf{q1}}_{avg}^w = \frac{1}{N}\sum_{m=1}^{N} \overline{\mathbf{q1}}_m^w, \quad \overline{\mathbf{q1}}_{max}^w = \max_{\text{along axis } m} \overline{\mathbf{q1}}^w, \tag{9}$$

$$\mathbf{q2}_{avg}^c = \frac{1}{N}\sum_{n=1}^{N} \mathbf{q2}_n^c, \quad \mathbf{q2}_{max}^c = \max_{\text{along axis } n} \mathbf{q2}^c, \quad \mathbf{q2}_{avg}^w = \frac{1}{N}\sum_{m=1}^{N} \mathbf{q2}_m^w, \quad \mathbf{q2}_{max}^w = \max_{\text{along axis } m} \mathbf{q2}^w, \tag{10}$$

$$\overline{\mathbf{q2}}_{avg}^c = \frac{1}{N}\sum_{n=1}^{N} \overline{\mathbf{q2}}_n^c, \quad \overline{\mathbf{q2}}_{max}^c = \max_{\text{along axis } n} \overline{\mathbf{q2}}^c, \quad \overline{\mathbf{q2}}_{avg}^w = \frac{1}{N}\sum_{m=1}^{N} \overline{\mathbf{q2}}_m^w, \quad \overline{\mathbf{q2}}_{max}^w = \max_{\text{along axis } m} \overline{\mathbf{q2}}^w. \tag{11}$$

Through the above Equations 8 - 11, we can employ average and max pooling to obtain high-order feature representations from the basic and correlation feature representations of different granularities for sentences $Q1$ and $Q2$. Conceptually speaking, the average and max pooling are able to extract a set of global and key features, respectively.

Using the outputs of the above pooling operations, we can now generate the final sentence-level representations. First of all, for sentence $Q1$, we generate its final feature representation by combining all its feature representations, as shown in Equation 12. Similarly, the feature representation of sentence $Q2$ is generated with Equation 13. Next, we can generate the multi-granularity correlation feature representations for sentences $Q1$ and $Q2$ respectively, as shown in Equation 14.

$$\mathbf{f}_1 = [\mathbf{q1}_{avg}^c; \mathbf{q1}_{avg}^w; \overline{\mathbf{q1}}_{avg}^c; \overline{\mathbf{q1}}_{avg}^w; \mathbf{q1}_{max}^c; \mathbf{q1}_{max}^w; \overline{\mathbf{q1}}_{max}^c; \overline{\mathbf{q1}}_{max}^w], \tag{12}$$

$$\mathbf{f}_2 = [\mathbf{q2}_{avg}^c; \mathbf{q2}_{avg}^w; \overline{\mathbf{q2}}_{avg}^c; \overline{\mathbf{q2}}_{avg}^w; \mathbf{q2}_{max}^c; \mathbf{q2}_{max}^w; \overline{\mathbf{q2}}_{max}^c; \overline{\mathbf{q2}}_{max}^w], \tag{13}$$

$$\mathbf{f}_3 = [\overline{\mathbf{q1}}^c_{avg}; \overline{\mathbf{q1}}^w_{avg}; \overline{\mathbf{q1}}^c_{max}; \overline{\mathbf{q1}}^w_{max}; \overline{\mathbf{q2}}^c_{avg}; \overline{\mathbf{q2}}^w_{avg}; \overline{\mathbf{q2}}^c_{max}; \overline{\mathbf{q2}}^w_{max}]. \tag{14}$$

In addition, we utilize the final semantic representations ($\mathbf{f}_1$ and $\mathbf{f}_2$) of sentences $Q1$ and $Q2$ to obtain their interactions, using the following operations:

$$\mathbf{ab} = |\mathbf{f}_1 - \mathbf{f}_2|, \quad \mathbf{mu} = \mathbf{f}_1 \times \mathbf{f}_2, \tag{15}$$

where $\times$ denotes the element-wise multiplication.

Finally, we concatenate these interactions to generate the final representation of multi-granularity correlation with Equation 16, which is transferred to the prediction layer.

$$\mathbf{F} = [\mathbf{ab}; \mathbf{mu}; \mathbf{f}_3]. \tag{16}$$

2.4 Prediction Layer

The prediction module is a multi-layer perceptron (MLP) classifier. It has three dense sub-layers, where the first two dense layers are activated with the ReLU function (Nair and Hinton, 2010) and the last dense layer is connected with the sigmoid activation function in our experiments.

2.5 Loss Function

For training, we utilize the modified binary cross-entropy loss (Su, 2017). In our notation, y_{true} is the actual label of a training sample and y_{pred} is the corresponding predicted label. For the convenience of comparison, the traditional binary cross-entropy is given in Equation 17:

$$L_{\text{old}} = -\sum(y_{\text{true}} \log y_{\text{pred}} + (1 - y_{\text{true}}) \log(1 - y_{\text{pred}})). \tag{17}$$

In order to improve its performance, the unit step function $\theta(x)$ (defined in Equation 18) and threshold m (set as 0.7) are introduced. The newly modified binary cross-entropy loss is defined in Equation 19. With this novel loss function, the model will be forced to focus on the indistinguishable training samples, which makes the classification perform better.

$$\theta(x) = \begin{cases} 1, x > 0 \\ \frac{1}{2}, x = 0. \\ 0, x < 0 \end{cases} \tag{18}$$

$$L_{\text{new}} = -\sum \lambda(y_{\text{true}}, y_{\text{pred}})(y_{\text{true}} \log y_{\text{pred}} + (1 - y_{\text{true}}) \log(1 - y_{\text{pred}})), \tag{19}$$

where $\lambda(y_{\text{true}}, y_{\text{pred}})$ is defined as:

$$\lambda(y_{\text{true}}, y_{\text{pred}}) = 1 - \theta(y_{\text{true}} - m)\theta(y_{\text{pred}} - m) - \theta(1 - m - y_{\text{true}})\theta(1 - m - y_{\text{pred}}). \tag{20}$$

3 Experiments and Results

3.1 Datasets

We conduct experiments on two Chinese sentence intention matching data sets, i.e., BQ and LCQMC. BQ is a Chinese bank question pair data set for sentence intention equivalence identification, which is a classic intention matching task (Chen et al., 2018). LCQMC is a generic corpus mainly for intention matching collected from Baidu Knows (Liu et al., 2018). The two datasets consist of a large set of instances in the form of ($Q1$, $Q2$, *Label*), where $Q1$ and $Q2$ are two Chinese sentences, and *Label* is the label indicating whether $Q1$ and $Q2$ share the same semantic intention. A summary of these data sets is provided in Table 3.

Table 3: Experimental data sets.

Dataset	Language	Source	Scale (train/valid/test)	pos:neg
LCQMC	Chinese	Baidu Knows	238,766/8,802/12,500	1.35:1
BQ	Chinese	WeBank	100,000/10,000/10,000	1:1

3.2 Parameter Settings

In our experiments, the embedding dimension is 300 in the multi-granularity embedding layer. The encoding dimension is set as 300 in the intra-correlation encoding layer. For the LSTM layer, dropout (Srivastava et al., 2014) rates of 0.5 and 0 are used for BQ and LCQMC respectively. In the BiLSTM layer, the dropout rates are set to 0.52 and 0.25 respectively for BQ and LCQMC. Dropout rate of 0.5 is used in the prediction component which actually consists of two densely connected hidden layers with 600 units in each layer and one classification output node with a sigmoid activation function. Adam with default parameters is adopted as the optimizer (Kingma and Ba, 2015). All the experiments are executed on a Thinkstation P910 workstation equipped with dual Xeon E5-2600 processors, 192 GB memory, and one Nvidia 2080Ti GPU.

3.3 Experimental Results

A comparison of our work with the baseline methods is shown in Table 4. We can observe that the performance of our model (ICE) is superior to all the compared methods in terms of most measures. There are three choices for pre-trained embeddings. The first one is word embedding with Word2Vec (Mikolov et al., 2013), the other is word embedding with GloVe (Pennington et al., 2014), and the last one is Glyce embedding which is first applied in Chinese tasks (Meng et al., 2019). The differences among them lie in that Word2Vec is a predictive model, GloVe is a count-based model, and Glyce generates glyph-vectors for Chinese character representations. Predictive models learn their embeddings in order to improve their predictive ability; count-based models learn their embeddings by essentially reducing dimension on the co-occurrence matrix; and Glyce, just like word embeddings, provides a general way to model character semantics in logographic languages. Similar to experiments using BiLSTM, BiMPM,

Table 4: Experimental results on LCQMC and BQ.

Task	Model	Precision	Recall	F_1-score	Accuracy
LCQMC	BiLSTM$_{word}$	67.4	91.0	77.5	73.5
	BiLSTM$_{char}$	70.6	89.3	78.9	76.1
	BiMPM$_{word}$	77.6	93.9	85.0	83.4
	BiMPM$_{char}$ (Chen et al., 2018)	77.7	93.5	84.9	83.3
	DFF$_{word}$	78.58	93.88	85.51	84.15
	DFF$_{char}$ (Zhang et al., 2019)	77.69	**94.08**	85.06	83.53
	MSEM (Huang et al., 2019)	78.90	93.73	85.68	-
	MGF (Zhang et al., 2020)	81.39	92.90	86.72	85.83
	BiMPM+Glyce (Meng et al., 2019)	80.4	93.4	86.4	85.3
	ICE	**83.34**	91.80	**87.33**	**86.73**
BQ	BiLSTM$_{word}$	75.04	70.46	72.68	73.51
	BiMPM$_{word}$ (Chen et al., 2018)	82.28	81.18	81.73	81.85
	DFF$_{word}$	84.43	77.48	80.70	81.59
	DFF$_{char}$ (Zhang et al., 2019)	85.38	76.33	80.52	81.69
	MSEM (Huang et al., 2019)	82.88	84.36	83.62	-
	MGF (Zhang et al., 2020)	**89.24**	74.67	81.21	82.86
	BiMPM+Glyce (Meng et al., 2019)	81.93	85.54	83.70	83.34
	ICE	84.06	**85.65**	**84.77**	**84.71**

DFF, and MGF, our model utilizes the pre-trained word embedding from Word2Vec. Compared with BiLSTM, BiMPM, and DFF, our model dramatically outperform them. This is probably because the three compared methods only consider one specific granularity, i.e., character or word, which is inadequate to capture enough features. Different from them, our model considers multi-granularity features to encode sentences, which can provide more effective information. In comparison with MSEM and MGF, our model performs better in terms of F_1-score and accuracy. Although MSEM and MGF consider concatenating word and character embeddings together to generate the final text representation, they do not capture the correlation features between different granularities, which leads to limited performance improvement. Besides, MSEM utilizes GloVe embeddings, which does not improve the performance. Compared with BIMPM+Glyce, even though Glyce has achieved good results on other Chinese language tasks, our model also outperforms it in current task.

3.4 Further Analysis

3.4.1 Comparison with BERT

Compared with BERT-based methods, our model performed comparably as reported in Table 5. BERT utilizes context information of characters to extract features, and dynamically adjusts embeddings of characters according to different contexts, which solves the polysemy problem suffering Word2Vec and thus helps achieve outstanding performances (Devlin et al., 2019). According to Table 5, our model surpasses BERT-based models on LCQMC and works comparably with them on BQ. This is probably because our model implements the intra-correlation encoding component, which enables us to capture sentence feature information from multiple perspectives and correlation information between sentences on different granularities.

Table 5: Comparison of our model with BERT-based methods.

Task	Metrics	Model	(M = Million)
LCQMC	Accuracy(#FLOPs)	BERT (Liu et al., 2020)	86.68(21785M)
	Accuracy(#FLOPs)	DistilBERT (6 layers)	84.12(10918M)
	Accuracy(#FLOPs)	DistilBERT (3 layers)	84.07(5428M)
	Accuracy(#FLOPs)	DistilBERT (1 layers)	71.34(1858M)
	Accuracy(#FLOPs)	FastBERT (speed=0.1)	86.59(12930M)
	Accuracy(#FLOPs)	FastBERT (speed=0.5)	84.05(6352M)
	Accuracy(#FLOPs)	FastBERT (speed=0.8)	77.45(3310M)
	Accuracy(#FLOPs)	**ICE**	**86.73(4.1M)**
BQ	Accuracy(#FLOPs)	BERT (Sun et al., 2020)	84.8(-)
	Accuracy(#FLOPs)	**ICE**	**84.71(2.5M)**

Moreover, in contrast to BERT-based approaches, our model is more concise and requires less computing power. We employ the #FLOPs to further evaluate the model. #FLOPs (number of floating-point operations) is a measure of the computational complexity of models, which indicates the number of floating-point operations that the model performs for a single process. Generally speaking, the bigger the model's #FLOPs is, the longer the inference time will be. With the same accuracy, models with lower #FLOPs are more efficient. When our graphics processing card resources are insufficient, ICE has much lower #FLOPs than BERT-based approaches but achieves comparable results.

3.4.2 Effectiveness of Modified Loss Function

In this section, we verify the effectiveness of the modified binary cross-entropy (BCE) loss. As shown in Table 6, compared with the traditional binary cross-entropy defined in Equation 17, the modified binary cross-entropy loss function in Equation 19 has achieved better performance.

This corroborates that the modified BCE loss function is more effective to allow the model to focus on the indistinguishable training samples, making the classification perform better.

Table 6: Experimental results using different loss functions.

Task	Model	Precision	Recall	F_1-score	Accuracy
BQ	ICE(BCE)	83.74	84.68	84.15	84.16
	ICE(Modified BCE)	**84.06**	**85.65**	**84.77**	**84.71**
LCQMC	ICE(BCE)	81.65	**92.35**	86.64	85.81
	ICE(Modified BCE)	**83.34**	91.80	**87.33**	**86.73**

4 Related Work

Sentence intention matching is critical for a series of downstream tasks, such as information retrieval, question answering, and machine translation.

With the development of deep neural networks, sophisticated models for SSM task have been evolving rapidly (Chen et al., 2017; Lai et al., 2019; Huang et al., 2019; Liu et al., 2020). The companion of attention mechanism with sequence models have achieved promising performance in machine translation (Bahdanau et al., 2015) and then is applied in many other tasks in natural language processing. It has also rendered encouraging effects in the SSM task (Wang et al., 2015; Wang et al., 2017; Tay et al., 2018; Duan et al., 2018; Kim et al., 2019). Wang et al. propose a multi-angle bidirectional attention mechanism in SSM task, and the effect of the model is remarkable (Wang et al., 2017). In addition, Sha et al. put forward an attention mechanism by repeating two sentences to improve the model's memory and obtain better textual semantic representation (Sha et al., 2016). Generally speaking, through using attention mechanisms, key feature representations in the text can be captured and benefit precise matching.

Although the above methods extract key feature representations with the attention mechanisms or introduce external syntactic information in the text sequences, these methods only achieve limited improvement. A number of researchers discover that the granularity of text is also crucial for capturing deep semantic features of the text. In particular, Huang et al. propose a word representation layer, which consists of word embedding and character representation, to capture multi-granularity feature representations (Huang et al., 2017). The acquisition and integration of text features at different granularities are considered in (Zhang et al., 2020) and achieve interesting results. However, their work simply integrates multi-granularity features, without taking into account the correlation of text features between different granularities.

The pre-trained language model BERT, which solves the polysemy problem of Word2Vec, has proven to be highly effective (Devlin et al., 2019). However, BERT is often computationally expensive in many practical scenarios. Thus, it is hard to be readily implemented with limited resources. Therefore, a series of smaller, faster, cheaper, and lighter of pre-trained BERT-based models emerge widely, such as DistilBERT (Sanh et al., 2019) and FastBERT (Liu et al., 2020). These models are optimized in terms of running speed and resource utilization, which inevitably reduces the effectiveness of the original BERT model. How to achieve comparable performance with BERT and require less computational resources is critical and urgent in model design for NLP applications.

In this paper, in order to better capture correlation features between different granularities, we propose an intra-correlation encoding framework for SIM task, which considers the correlation between text features from character-granularity and word-granularity. With less requirement on computational resources, our proposed model can achieve better or comparable performance with the state-of-the-art BERT.

5 Conclusions

For sentence intention matching tasks, we propose a novel method, named the intra-correlation encoding model. It combines character-granularity and word-granularity features to model sentence intention, and utilizes soft alignment attention to enhance the local information of sentences on the different levels. It can capture sentence feature information from multiple perspectives and correlation information between different levels of sentences. Experiments on two datasets demonstrate that our model outperforms non-BERT-based models and achieves at least comparable accuracy with BERT-based models, but runs much more efficiently than BERT. In the future, we would attempt to join multi-granularity embeddings and BERT together, so as to further improve the performance. The generalization of our model to other languages will also be investigated.

Acknowledgements

The research work is partly supported by National Key R&D Program of China under Grant No.2018YFC0831700 and No.2018YFC0830705, National Natural Science Foundation of China under Grant No.61502259, and Key Program of Science and Technology of Shandong under Grant No.2019JZZY020124.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1657–1668.

Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou Tang. 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.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Chaoqun Duan, Lei Cui, Xinchi Chen, Furu Wei, Conghui Zhu, and Tiejun Zhao. 2018. Attention-fused deep matching network for natural language inference. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4033–4040.

Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural language inference over interaction space. In *Proceedings of the International Conference on Learning Representations*.

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.

Qiang Huang, Jianhui Bu, Weijian Xie, Shengwen Yang, Weijia Wu, and Liping Liu. 2019. Multi-task sentence encoding model for semantic retrieval in question answering systems. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1–8.

Seonhoon Kim, Inho Kang, and Nojun Kwak. 2019. Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6586–6593.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*.

Yuxuan Lai, Yansong Feng, Xiaohan Yu, Zheng Wang, Kun Xu, and Dongyan Zhao. 2019. Lattice CNNs for matching based Chinese question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6634–6641.

Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. 2019. Is word segmentation necessary for deep learning of Chinese representations? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3242–3252.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. LCQMC: A large-scale Chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Haotang Deng, and Qi Ju. 2020. FastBERT: A self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for Chinese character representations. In *Advances in Neural Information Processing Systems*, pages 2742–2753.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 807–814.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of the 33th Conference on Neural Information Processing System*.

Lei Sha, Baobao Chang, Zhifang Sui, and Sujian Li. 2016. Reading and thinking: Re-read LSTM unit for textual entailment recognition. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 2870–2879.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Jianlin Su. 2017. Text emotion classification (IV): Better loss function. Web page. `https://spaces.ac.cn/archives/4293`.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Chuanqi Tan, Furu Wei, Wenhui Wang, Weifeng Lv, and Ming Zhou. 2018. Multiway attention networks for modeling sentence pairs. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4411–4417.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Hermitian co-attention networks for text matching in asymmetrical domains. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4425–4431.

Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2015. Syntax-based deep matching of short texts. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1354–1361.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4144–4150.

Xu Zhang, Wenpeng Lu, Fangfang Li, Xueping Peng, and Ruoyu Zhang. 2019. Deep feature fusion model for sentence semantic matching. *Computers, Materials & Continua*, 61(2):601–616.

Xu Zhang, Wenpeng Lu, Guoqiang Zhang, Fangfang Li, and Shoujin Wang. 2020. Chinese sentence semantic matching based on multi-granularity fusion model. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 246–257.