Journal of Computers Vol. 31 No. 6, 2020, pp. 78-90
doi:10.3966/199115992020123106007

Integration of Multi-granularity Information for Natural Language Inference

Shu-Yu Cheng[1*], Ze-Ying Guo[2], Jian Yin[2]

[1] Anhui Vocational College of Electronics & Information Technology, Bengbu 233060, Anhui, China
csygold@163.com

[2] School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 510275, Guangdong, China
michellegzy@163.com, issjyin@zsu.edu.cn

Received 13 August 2019; Revised 17 January 2020; Accepted 2 March 2020

Abstract. Research on natural language inference is an important task in the field of natural language processing. Traditional methods mainly rely on feature engineering, external semantic resource and tools, and the machine learning methods are combined to complete the classification of text entailment relationship. Existing deep learning methods mainly utilize deep neural network to model the sentence sequence in order to complete the representation and matching of sentence, but the following problems still exist: (1) The sentence feature representation is not rich enough; (2) The semantic expression of low-frequency words by using word vector is insufficient; (3) The problem of interactive information between sentences is ignored during modeling of sentence pair. In order to address the above three problems, from the perspective of the multi-granularity of character, word and sentence, we propose the natural language inference model with information fusion and interaction between character & word and word & sentence, and utilize deep neural network (CNN-BiLSTM) to complete the classification of text entailment relationship. Extensive experiments were conducted on the two public datasets of SNLI and MNLI. With less parameters, our model outperforms the state-of-the-art models.

Keywords: attention mechanism, matching strategy, multi-granularity, NLI, representation learning

1 Introduction

Natural Language Inference (NLI) is defined as the inference relation between sentence pair directly, and as a fundamental part of natural language processing, it belongs to natural language understanding. The basic task of NLI is to determine whether the semantics of hypothesis sentence (H) can be inferred from the semantics of premise sentence (P). If the semantics of sentence H can be inferred from the semantics of sentence P, then there is entailment relation between sentence pair P-H, as shown in Table 1.

Table 1. Data sample of NLI task

Example	Relationship
P: Chinese first-tier cities are Beijing, Shanghai, Guangzhou, and Shenzhen. *H:* Guangzhou is one of Chinese first-tier cities.	Entailment
P: "Oriental Moscow" Harbin, obtained "The National Civilized City" title. *H:* Moscow obtained "The National Civilized City" title.	Contradiction
P: Tencent is a large internet company. H: Microsoft is the largest software company.	Neutral

NLI is a classification problem in form, with sentence pair as the classification object. Its most basic task is to determine whether there is entailment relation between sentences; for non-entailment relation,

* Corresponding Author

further determine whether it is contradictory or neutral relation. In this way, the three-classification task can be completed.

The traditional natural language inference adopts the method based on feature classification, and this method determines whether there is entailment relation between the precondition sentence and hypothetical sentence by calculating the similarity between the two sentences [1-3]. The calculation of similarity between sentences depends on the extraction of sentence features [4], which is supplemented with external semantic resources [5].This method generally requires high time and computational cost, and the results have weak reproduction and transfer learning ability. In the meantime, this method does not conduct much research on the semantics contained in the sentence, and the main reason is that the traditional method cannot learn large-scale corpus or mine semantic association.

In recent years, with the continuous deepening of researches on deep learning in the field of natural language processing, significant progress has been achieved in utilizing the deep learning method to study natural language inference. The deep learning method can automatically learn related features of sentence by building deep and complicated neutral network structure. The deep learning-based NLI method inputs the precondition sentence and hypothetical sentence into the deep neutral network, and utilizes the deep neutral network to model the sentence sequences to complete the classification task.

Current mainstream research works mainly utilize deep neural network to model the sentence sequence to complete the presentation and matching of sentences. Although certain progress has been achieved, there are still many aspects that require improvement. In the deep neural network, by mapping the sentence to the vector space, it will be easier to mine the computation of sentence features and the association between sentence features, and in this way, we can better learn the sentence features. However, the following problems still exist in the modeling and relation inference of sentence pair:

(1) In the training corpus, the semantic expressions of low-frequency words and unregistered words are insufficient, and the semantic information of word vector itself is insufficient. As a result, the representation of sentence has low quality, which reduces the model performance.

(2) Current methods ignore the interaction information between sentences, and it can't effectively judge the inference relation of sentence pairs.

(3) The popular models can't take into account the global and local characteristics of the sentence, which causes the missing of sentence semantic information and the misinterpretation on entailment and neutral relations.

In view of aforementioned three problems, we proposed a multi-granularity model with information interaction for NLI. The main contributions of this paper can be summarized as follows:

(1) In view of low-frequency words and unregistered words, we propose a method of integrated character and word information. By constructing a character-level convolutional neural network (Character CNN) model, it can reduce the input granularity, and combine the original word embedding in corpus. Besides, we use two granularity features of characters and words, to increase the quality of sentence expression and improve the judgment effectiveness on neutral relationship of the model.

(2) In view of interaction information between sentences, we propose a method to learn the interaction information between sentences. Considering sentence matching, we use the bidirectional long short memory neural network model (BiLSTM) to collect the context information during sentence coding. By adopting different levels and multi-granularity matching strategy and integrated attention mechanism in modeling interactive relationship between words and sentences, it can reduce the missing of semantic information and increase the recognition accuracy of entailment relationship and contradictory semantics.

(3) In view of characteristics of word formation and sentence structure, we proposed a hierarchical structure combining the convolution neural network model and the bidirectional long short memory network (CNN-BLSTM). Based on the structure, we integrate aforementioned two methods to further increase the quality of the model.

The rest of this paper is organized as follows. In Section 2, related work is introduced. In Section 3, we propose a multi-granularity natural language inference model for information integration and interaction, and this model adopts the character-letter information integration method for representation learning; as for text matching, the different-level and multi-granularity sentence interaction strategy is employed. In Section 4, we conduct specific comparison experiment on standard evaluation dataset by using the method adopted in the model. In Section 5, we compare the performance of our model with that of state-of-the-art models via experiment. In Section 6, we made further analysis of the experimental results. Finally, we draw conclusion in Section 7.

2 Related Work

NLI includes two methods which are based on feature classification and deep learning. The method based on feature classification utilizes artificial statistics and tag to complete extraction of sentence features, such as the bag-of-words model [5], N-gram and TF-IDF method. The deep learning-based method automatically learns related features of sentence by building deep and complicated neutral network structure, including the representation learning method and text matching method.

The representation learning method utilizes the context information of sentence for modeling of sentence, and encodes the sentence into fixed dimension vector; then, the vector is matched, and the obtained matching vector is input into the subsequent neutral network to complete the classification task. The simplest method to obtain sentence vector is to combine the word embedding of all words in a sentence. The word embedding uses a vector with fixed dimension to represent the word, so that it can cover the semantic information of word and reflect their association.

The collection of word vector is generally based on the co-occurrence statistical method or similar context method. Now most NLI tasks will directly introduce word pre-training vectors such as word2vec [6], glove [7] to help the model get more semantic information, but this method has several defects, such as rare words, new words, unregistered words, the antonyms of similar context and morphing of words, and as a result, the current methods cannot to meet the needs of NLI task.

The common representation learning methods include the methods based on the RBM model, RNN model, LSTM model, BiLSTM model and CNN model, respectively. The RBM model [8] is the earliest deep leaning model which utilizes the reconstruction error as the feature to determine the entailment relation between the sentence pair. The RNN model [9] keeps the sequence information based on the input sequence of sentence, but for processing of long sentence, it may result in loss of historical information due to error propagation and update computation. The LSTM model [10-11] has introduced the “gate” mechanism based on the RNN model, so that the hidden layer can store the historical information longer. However, the coding can only be conducted according to the context information which has been read, and the other context information cannot be utilized. The BiLSTM model [12] combines the LSTM model of positive and inverted sequences, which can help extract the global information of sentence without being affected by the sentence sequence, but it fails to consider the local features of sentence. The CNN model [13] has the characteristic to extract local features, which can also maintain the relative location of other features, but it cannot utilize the semantic information contained in the sentence sequence.

All the above methods have the defect of not being able to simultaneously consider the distribution of global features and positioning of local features of sentence during the sentence modeling, which causes semantic loss. In the meantime, the major difficulty in identification of the entailment relation between sentences is to distinguish the entailment relation from the neutral relation. The popular methods based on the similarity between sentences do not possess the semantic inference ability, and similarity is not equivalent to entailment.

The text matching method does not need to generate sentence vector, which directly aggregates the matching vector of sentence, calculates the probability and predicts the classification. Generally, this kind of method would introduce the attention mechanism, including the word-by-word attention [11] mechanism for sentence-level matching and mLSTM [14] for word-level matching. Such single-granularity matching method does not explore much of the interactive information between sentence pair, which has ignored the interactive features between sentences.

Current mainstream NLI research works mainly utilize deep neural network to complete the representation and matching of sentence. How to effectively utilize the generated sentence vector, fully analyze the semantic information contained in the sentence vector and extract the semantic relation between two sentence vectors is critical to text matching. Wang et al. [9] proposed a computational method for sentence matching from multiple perspectives, which can extract the sentence features from different perspectives. By encoding two given sentences respectively and matching from various directions, the semantic information of sentence can be fully extracted.

By referring to the multi-perspective matching method, this paper studies how to combine and enrich the information of sentence itself during the modeling process of deep neutral network, and the attention mechanism is introduced to optimize the model. This work proposes two approaches for improvement: one approach aims to improve the representation learning quality through character granularity; the second approach tries to enhance the information interaction between sentences with different matching strategies, and the attention mechanism is integrated to improve the performance.

3 Model Design

In this paper, we proposed a model of multi-granularity information integration between characters and words, words and sentences for NLI tasks. The basic model is based on the hierarchical structure of CNN-BILSTM. Given a pair of sentences P and Q, the model estimates the probability Pr(y|P,Q)T through the following five layers. The overall architecture is shown in Fig. 1. We will introduce this model in the following subsection.

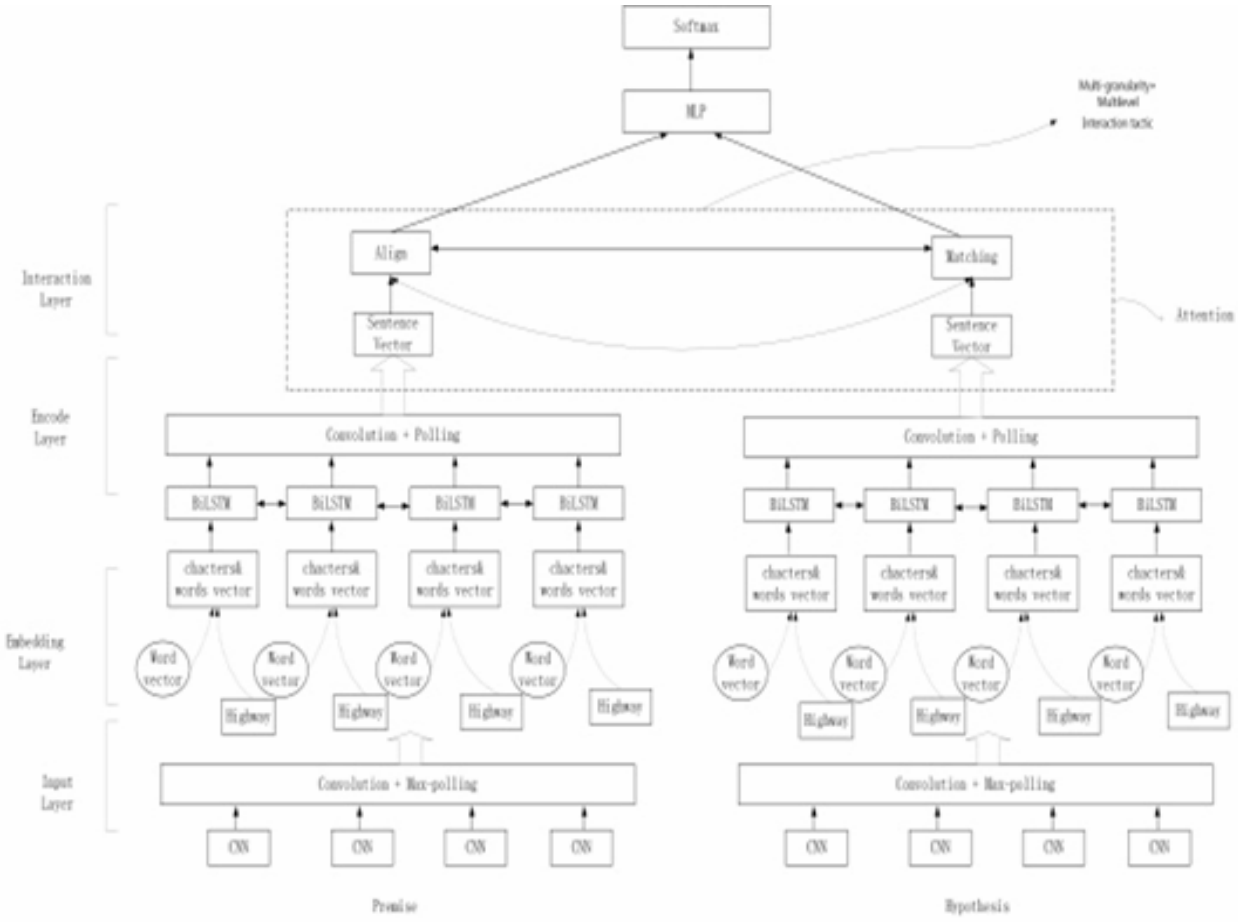


Fig. 1. Multi-granularity sentence interaction natural language inference model architecture

On the input layer, by taking character of word as input, we use convolutional neural network to obtain two granularity features of character and word for premise sentence P and hypothetical sentence H respectively, and combine with the convolutional and max-pooling operation to build the input for the embedding layer. On the embedding layer, we propose a method integrating character and word, feeding the character-composed embedding to the Highway network, and connect it with the pre-trained word embedding to generate a new word vector. On the encoding layer, we utilize the BiLSTM to encode the sequence of input word vector and generate the sentence representations, and through the convolutional and pooling operation, the sentence representations is input to the interaction layer for sentence matching. On the interaction layer, we propose the multi-granularity sentence interaction strategy. We utilize a BiLSTM to encode contextual embedding and adopt the multi-granularity matching strategy for

modeling of word and sentence with attention mechanism. Finally, the two sequences of matching vectors are aggregated on the integration layer (MLP), and the softmax function is applied for classification; the tag is entailment, contradiction or neutral.

The objective function of model as shown in Formula (1), where, N is the sample quantity, C is the tag type, is prediction tag, and $\hat{y}$ is the distribution of actual tag y .

$$\mathrm{L}\left(y,\hat{y}\right)=-\frac{1}{N}\sum_{n\in N}\sum_{i\in C}y_{n,i}\log\hat{y}_{n,i}. \tag{1}$$

3.1 Input Layer

The goal of this layer is to utilize the CNN to extract characters combing features under different convolution kernels for each word, and by learning the input feature vector, the feature mapping on different levels can be obtained through convolutional operation of the matrix representation of word. The computation method is expressed as follows:

$$\left(f^k\left[i\right]\right)_H=\tanh(\left\langle C^k\left[*,i:i+w-1\right],H\right\rangle)+b). \tag{2}$$

$$y_H^k=\max_i(f^k\left[i\right])_H. \tag{3}$$

where $C^k\left[*,i:i+w-1\right]$ is the real value from i^{th} column to $(i+w-1)^{th}$ column in the word matrix; H is the convolution kernel, and y_H^k is the feature value of word k under convolution kernel. The character CNN model can not only improve the quality of sentence representation, but also increase the performance judgment of the model in neutral relation.

3.2 Word Embedding Layer

In order to reduce the loss of word semantic information generated by the character-level CNN model, we propose the character-word fusion method. Transfer the character-level word from the input layer to the Highway network to conduct nonlinear transformation, and obtain the word vector v_c ; then, connect v_c with the pre-trained word vector v_w to obtain the word vector $v_{combined}$ that combines the two-granularity information of character and word. The computation method is expressed as follows:

$$v_{combined}=\left[v_c,v_w\right]. \tag{4}$$

3.3 Coding Layer

The main task of this layer is to utilize the BiLSTM model to conduct coding of word vector transferred from previous layer and generate the sentence representation vector. The BiLSTM model inputs code in two directions, which can capture the overall semantics of context and better record the information of complete input sequence. In the BiLSTM model, the word vector *vocombined* is the input, and pooling and connection operation is conducted to the output at each moment. The computation method is expressed as follows:

$$y_{\text{mean}}^P=\sum_{t=1}^T\frac{y_t^P}{T},y_{\max}^P=\max_{1\le t\le T}y_t^P. \tag{5}$$

$$y_{mean}^H=\sum_{t=1}^T\frac{y_t^H}{T},y_{\max}^H=\max_{1\le t\le T}y_t^H. \tag{6}$$

$$s^P=\left[y_{mean}^P,y_{\max}^P\right],s^H=\left[y_{mean}^H,y_{\max}^H\right]. \tag{7}$$

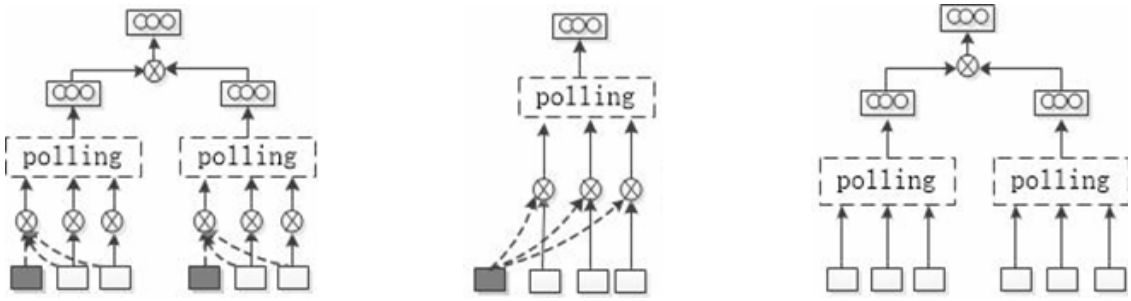
where, y_t^P and y_t^H are the context vectors for each time-step of sentences P and H. Average pooling considers the local information on each aspect, which can prevent loss of information; max-pooling only keeps the biggest feature value within given scope, which can help strengthen important semantics.

3.4 Interaction Layer

The purpose of this layer is matching the two sentences P and H. Sentence matching consists of comparing two sentences and determining their relation, and its main task is to aggregate the combination features between P and H. The traditional matching method mainly conducts vector comparison after vector representation of sentence pair [10], or directly connects the precondition text and hypothetic text as the input for a single sentence [11]. Some researchers obtained the interactive information between sentence vectors through matrix [15]; some introduced the attention mechanism for consideration [13], and they all achieved great performance at current stage. However, all these methods adopt single-granularity word-by-word or sentence-by-sentence interactive matching. Such matching method only focuses on capturing the semantic information of sentence itself, while ignoring the combination features between words within the sentence and between sentences on different levels, which result in loss of semantics. In this section, we mainly consider the direction feature involved in the inference relation and the multi-granularity information possessed by the sentence. By integrating the attention mechanism, we propose a multi-granularity and different-level sentence interaction matching method based on the BiLSTM model. The attention weight is used as the interactive information between the precondition sentence and hypothetical sentence. Richer semantic combination features can be obtained through different interactive strategies, such as single-granularity interaction on the same level and cross-level multi-granularity interaction. Through weighting with context vector, the new expression vector of sentence is output based on pooling.

3.4.1 Single-granularity Interaction

It refers to the interaction between word granularities or sentence granularities. By conducting full matching of context vector generated during the modeling process of sentence pair, the interactive features between words can be extracted; we define three interaction strategies for the interaction between word granularities of different sentences and the interaction between sentence granularities of different sentences, respectively, as shown in Fig. 2.



(a)Inner-word-by-word (b) Outer-word-by-word (c) Different-sentence-by-sentence

Fig. 2. Single-granularity interaction, where in the parallel network, the most left grey rectangle refers to the contest vector of this sentence at a certain moment; all right rectangles represent the context vector of this sentence at any moment, the same below

(1) *Inner-word-by-word Interaction*

As shown in Fig. 2(a), inner-word-by-word interaction in the same sentence will calculate the attention weight between the context vectors of sentence itself. The computation method is as shown in Formulas8-10, where, T is the sequence length. This method can capture the semantic features within the sentence, and highlight the importance of each word in the sentence.

$$m_{ij}^P = f_m\left(y_i^P, y_j^P\right),\ m_{ij}^H = f_m\left(y_i^H, y_j^H\right). \quad \textbf{(8)}$$

$$m'^{P}_{i} := \sum_{t=1}^{T} \frac{\exp\left(m^{P}_{it}\right)}{\sum_{k=1}^{T}\exp\left(m^{P}_{ik}\right)} y^{P}_{t}. \tag{9}$$

$$m'^{H}_{i} := \sum_{t=1}^{T} \frac{\exp\left(m^{H}_{it}\right)}{\sum_{k=1}^{T}\exp\left(m^{H}_{ik}\right)} y^{H}_{t}. \tag{10}$$

(2) *Outer-word-by-word Interaction*

As shown in Fig. 2(b), in this strategy, we will complete the calculation of cross attention weight by aligning the context vectors contained in P with the vectors in H one to one. The computation method is as shown in Formulas11-12, where, s_{ij} is the element in alignment matrix $S_{P\times H}$, while M and N are the lengths of sentences P and H respectively; α_i is the alignment of the context vector of sentence P to y^{H}_{i}, and β_j is the alignment of the context vector of sentence H to y^{P}_{i} .This method can capture the importance of each field of P in H.

$$S_{P\times H} = \left(y^{P}\right)^{T} \cdot \left(y^{H}\right), s_{ij} = f_{m}\left(y^{P}_{i}, y^{H}_{j}\right). \tag{11}$$

$$\alpha_i := \sum_{j=1}^{M} \frac{\exp\left(s_{ij}\right)}{\sum_{k=1}^{M}\exp\left(s_{ik}\right)} y^{P}_{i}, \beta_j := \sum_{i=1}^{N} \frac{\exp\left(s_{ij}\right)}{\exp\left(s_{kj}\right)} y^{H}_{i}. \tag{12}$$

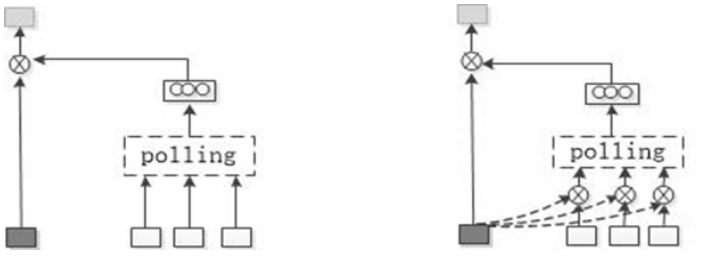
(3) *Different-sentence-by-sentence Interaction*

As shown in Fig. 2(c), in this strategy, we will directly calculate the representation vectors of two sentences after modeling of BiLSTM layer. This method can obtain the sentence-level interactive information, and the contribution to the semantic relation inference of sentence pair. The computation method is as shown in Formula 13. Max-pooling is used to select the feature value.

$$v = \left(m\right)_{max\text{-}polling} = \max\left[y^{P} \odot y^{H}\right]. \tag{13}$$

3.4.2 Multi-granularity Interaction

Multi-granularity interaction is the interaction between word-level and sentence-level. The context vectors of sentence and word are matched to extract cross-level word and sentence feature. In this strategy, we define two methods of word-by-sentence interaction and attentive interaction, as shown in Fig. 3. This strategy has not only enriched the semantic information in sentence modeling, but also strengthened the inference of positive entailment in natural inference task.



(a) Word-by-Sentence (b) Attentive-Matching

Fig. 3. Multi-granularity interaction

(1) *Word-by-sentence Interaction*

As shown in Fig. 3(a), in this strategy, we will compare each context vector of sentence P with the representation vector of sentence H. The computation method is as shown in Formula 14. This method has integrated the matching between sentence vector and word-level context vector, which can obtain the

semantic similarity between each word in sentence P and each word in sentence H.

$$\left(m_t\right)_{full} = f_m\left(y_t^P, y^H\right). \tag{14}$$

(2) *Average-pooling-attentive-matching*

As shown in Fig. 3(b), in this strategy, we first calculate the matching value between each time-step contextual vector; then, we take it as the weight and calculate the representation vector of H by weighted averaging all the contextual embedding of H. The computation method is as shown in Formulas 15-16.

$$s_{i,j} = f_m\left(y_i^P, y_j^H\right), i, j \in \left(1, \ldots L\right). \tag{15}$$

$$\alpha_i = \frac{\sum_{j=1}^{T} s_{ij} \times y_j^H}{\sum_{j=1}^{T} s_{ij}}, \left(m_t\right)_{attention} = f_m\left(y_t^P, \alpha_t\right). \tag{16}$$

This method can obtain the cross granularity information of sentences P and H. In the meantime, the attention mechanism can be used to conduct feature selection.

(3) *Max-pooling-attentive–matching*

Max-pooling matching that integrates attention is as shown in Fig. 3(b). For the complete interaction process, refer to the average pooling matching, and replace the averaging operation with maximization operation. The computation method is as shown in Formula 17.

$$\alpha_i = \max\left(s_{ij} \times y_j^H\right), \left(m_t\right)_{\max-\text{att}} = f_m\left(y_t^P, \alpha_t\right). \tag{17}$$

3.4.3 Matching Function

In essence, the matching between sentences is the distance computation of sentence pair vector. There are several computation methods for distance function. By referring to the measurement method mentioned by Wang et al. [14] in the NLP task, we tried to conduct computation by using the following six different matching functions: Cosine, Euclidean distance + Cosine sum, Feedforward Neural Network (FNN), Element-wise Subtraction, Element-wise Multiplication and Element-wise Subtraction + Multiplication. See Table 2, where, v_1 and v_2 are two *d*–dimensional vectors.

Table 2. Matching function

Matching function	Corresponding formula
Cosine	$f_m(v_1, v_2) = cos(v_1, v_2)$
Euclidean + Cosine	$f_m(v_1, v_2) = \begin{bmatrix} \lVert v_1 - v_2 \rVert_2 \\ cos(v_1, v_2) \end{bmatrix}$
Feedforward Neural Network	$f_m(v_1, v_2) = F(v_1, v_2) = W(v_1, v_2) + b$
Element-wise Subtraction	$f_m(v_1, v_2) = (v_1 - v_2) \odot (v_1 - v_2)$
Element-wise Multiplication	$f_m(v_1, v_2) = v_1 \odot v_2$
Element-wise Subtraction + Multiplication	$f_m(v_1, v_2) = \begin{bmatrix} (v_1 - v_2) \odot (v_1 - v_2) \\ v_1 \odot v_2 \end{bmatrix}$

4 Experiments

In this section, we first introduce and compare related evaluation datasets for the NLI task. Then, we compare our model with state-of-the-art models on some standard evaluation datasets to analyze the effectiveness of our scheme.

4.1 Experiment Settings

Dataset: We conducted our experiments on Stan-ford Natural Language Inference (SNLI [10]) dataset and the Multi-Genre NLI (Multi-NLI [16]) dataset. The SNLI dataset consists of 570,000 sentence pairs written by humans, and each sentence pair includes precondition text, hypothetical text, tag, and five manual annotations; the Multi-NLI dataset consists of 433,000 sentence pairs written by humans, including multiple text sources such as Fiction, Government, Characters, 9/11, Telephone and Face-to-Face.

We conducted statistical analysis of two experimental datasets respectively in Fig. 4; we can see that both SNLI dataset and MNLI dataset have balanced tag distribution, mainly because both datasets were manually annotated. Therefore, during the experiment process, we did not need to consider the weights of different tags.

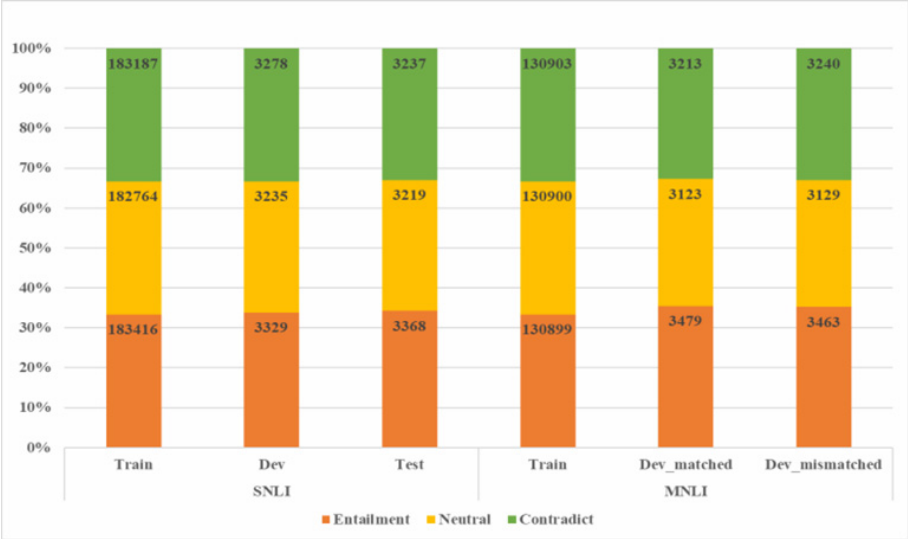


Fig. 4. Tag distributions of SNLI dataset and MNLI dataset samples

During the experimental evaluation in MNLI dataset, we set two different tasks: the evaluation in standard field was completed with the Dev_matched dataset; the cross-field evaluation was completed with the Dev_mismatched dataset. In this paper, we adopt cross verification with the MNLI dataset, 20% of dataset was used as the verification dataset, and the Dev_matched and Dev_mismatched datasets were used as the test datasets. We use accuracy as evaluation standard, which is shown as follows:

$$Accuracy = \frac{1}{|pairs|}\sum_{i} 1\left[\widehat{y_i} = y_i\right]. \tag{18}$$

where |pairs| is the number of sentence pairs; $\widehat{y_i}$ is the prediction tag of entailment relation; y_i is the actual tag; 1[·] is the representation function, the value is 1 when the decision condition is true, or otherwise, it is 0.

Parameter: The experimental parameter settings are shown in Table 3.

Table 3. Parameter Settings

Parameter	Settings
Learning Rate	0.0001
Batch Size	{32, 64, 128}
Convolution Kernel' Width(w)	[1, 2, 3, 4, 5, 6, 7]
Convolution Kernel' Size	[min {200, 50·w}]
Word Embedding	GloVe
Word Embedding	300
Character vector dimension	15
Maximum word length	15

Table 3. Parameter Settings (continue)

Parameter	Settings
Maximum sentence length	40
Highway layer dimension Highway	300
BiLSTM layer dimension	300
BiLSTM layer regular way	l_2
Dropout	0.3
Early stopping	5
Maximum number of iterations	30
optimization method	Adam

4.2 Methods for Comparison

We compare the performance of our model with that of some state-of-the-art models on SNLI dataset and MNLI dataset. For the SNLI dataset, the evaluation was based on the performance on training set and test set; for the MNLI dataset, the evaluation was based on the two datasets for standard evaluation (matched) and cross-field evaluation (matched).

BiMPM [17]: Extract the sentence features from multiple perspectives, and by encoding the two given sentences, conduct matching from multiple directions, which can fully extract the semantic information of sentence. According to Table4, we can see after optimization, the single BiMPM model could achieve an accuracy of 87.5%, and the combinational model could achieve an accuracy of 88.8%, which is a model with outstanding performance on the NLI task. However, because this model has integrated excessive dimension information, it might generate redundancy to a certain extent.

ESIM model [18]: Intensify the sequence information. By combining the Tree-LSTM model, the grammar analysis information of sentence can be integrated to better recognize the sentence structure, which has better performance in sentence modeling.

5 Results

Table 4 and Table 5 list the experimental results of some state-of-the-art models on the SNLI dataset and MNLI dataset in recent years, as well as the results of model proposed in this paper on corresponding dataset. We focused on comparing the optimal BiMPM model and ESIM model.

Table 4. Comparison results on SNLI dataset

Source	Model	Parameter number	Train Acc (%)	Test Acc (%)
Chen	ESIM	4.3m	76.3	75.8
Chen	ESIM + TreeLSTM (Ensemble)	7.7m	77.8	77.0
	Our Model	3.1m	78.0	77.2

Table 5. Comparison results on MNLI dataset

Source	Model	Parameter number	Train Acc (%)	Test Acc (%)
Bowman [19]	LSTM	3.0m	83.9	80.6
Mou [20]	Tree-based CNN	3.5m	83.3	82.1
Wang [14]	mLSTM + Attention	1.9m	92.0	86.1
Cheng [21]	LSTMN + Attention	3.4m	88.5	86.3
Parikh [22]	Decomposable Attention Model	582k	90.5	86.8
Wang [17]	BiMPM	1.6m	90.9	87.5
Chen [18]	ESIM	4.3m	92.6	88.0
Chen	ESIM + TreeLSTM (Ensemble)	7.7m	93.5	88.6
Wang	BiMPM (Ensemble)	6.4m	93.2	88.8
	Our Model	3.1m	91.7	88.5

From Table 5, it can be seen the ESIM model could provide an accuracy of 88.0%, and the accuracy of combinational model also reached 88.6%, which is also an outstanding model. However, because the tree

structure involves more parameters, this model has higher training consumption.

According to the results in above table, it can be seen that on the SNLI dataset, the model proposed in this paper could reach the accuracy of 88.5%, which was slightly lower than the accuracy of ESIM and ESIM + TreeLSTM model. The main reason is that in the combinational model, the advantages of different models are combined through the method of "voting", which has balanced both the advantages and shortages in model training. In the meantime, richer semantic information is introduced as support, so it has more advantages in sentence modeling. However, the combinational model generally requires more parameters for fine adjustment of model. For example, the parameters of BiMPM (Ensemble) model were 6.4m, and the parameters of ESIM (Ensemble) model reached 7.7m, while the parameters of the model proposed in this paper were only 3.1m. Our model has lighter architecture, which is more beneficial to training and optimization. On the MNLI dataset, our model outperformed the ESIM and ESIM + TreeLSTM combinational model, and the parameters of 3.1m were more beneficial to training and optimization.

6 Further Analyses

In order to further verify the generalization ability of the model proposed in this paper, by referring to the verification method adopted by Parikh et al. [15] on the SNLI dataset, we conducted experiment on the samples with three different entailment relations, and the experimental results are as shown in Fig. 5.

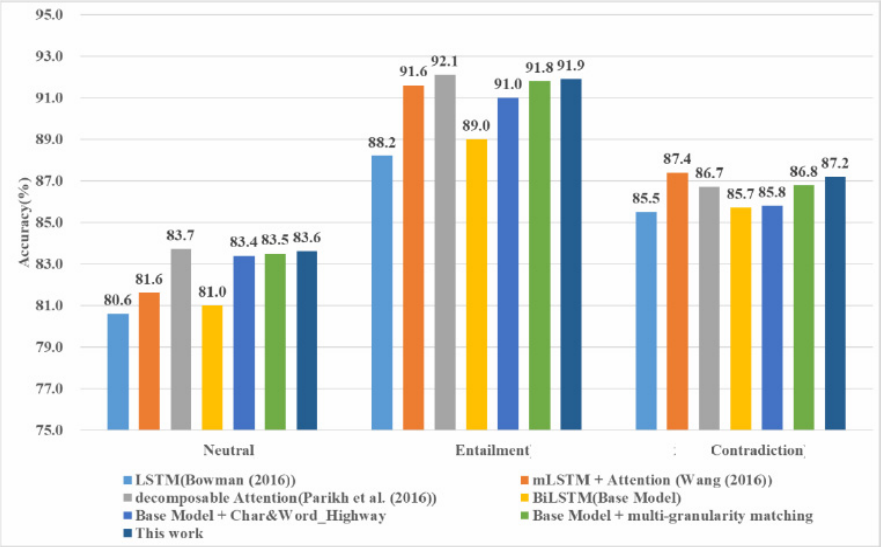


Fig. 5. Comparison of experimental results for certain tags on dataset

In which, the mLSTM model [14] connects the attention vectors generated by the two LSTM models modeled from the precondition sentence P and hypothetical sentence H, and then conducts matching and prediction. This method can keep the mismatching information during corresponding matching process. The less mismatching information is there, the more probable that the sentence relation is entailment, or otherwise, it will be determined as contradiction or neutral. This method which focuses on mismatching features can smartly ignore the high-frequency words with low recognition, such as common stop words. Therefore, this model has great performance in representation of contradicted semantics, and the accuracy could reach 87.4%. However, because it is over sensitive to the matching information, it has poor performance in recognition of neutral relation, and the accuracy is only 81.6%.

The Decomposable Attention model [22] conducts corresponding matching between each word in the precondition sentence and each word hypothetical sentence, and the neural network matrix operation is combined to decompose the attention mechanism solution of two sentences into two sub-problems. With this decomposition method, the attention operations of two sentences are parallel and independent, and it prefers words with similar semantics in weight assignment. Therefore, it has the best performance in determining the entailment relation with an accuracy of 92.1%. But it has poor performance in recognition of contradiction relation, with an accuracy of only 86.7%.

From Fig. 5, we can see that the model proposed in this paper could achieve the accuracies of 91.9%, 87.2% and 83.6% for recognition of entailment relation, contradiction relation and neural relation, respectively. It can be seen that after fusion with the character-word information, the model's performance in determination of neural relation can be improved in a certain degree; by introducing various different interactive strategies, it can help improve the recognition accuracies of entailment relation and contradiction semantics. In short, compared with other representative mainstream models, the model proposed in this paper can not only ensure effective recognition of entailment relation, but also ensure the recognition precision of contradiction relation, which has reduced the sensitivity in recognition of a certain relation. Its overall performance is better than the above models.

7 Conclusion

In this paper, first of all, for the low-frequency words, unregistered words and other problems in the NLI task, we proposed the CNN structure with character-level input to obtain the character information of word. By combining the pre-trained word vector as the new word representation, the feature information of the two granularities of character and word can be obtained at the same time. This method has enriched the semantic information of word, which can better help determine the entailment relation in the NLI task, and we also verified its effectiveness through experiment comparison and analysis. Secondly, for the interactive information between sentences during sentence modeling, the attention mechanism is integrated to propose 6 different interactive strategies. Through feature combination of different granularities, different levels and different pooling methods of words and sentences and the experimental comparison, we verified that the diversity of combinational features can improve the performance of NLI model. Finally, based on fusion of these two methods in the convolutional and bi-lateral long short-term memory neural network (CNN-BiLSTM), we proposed the natural language inference model with multi-granularity information interaction, and verified that this model had high accuracy and light structure in the NLI task through experiment. Furthermore, the model proposed in this paper could achieve the accuracies of 91.9%, 87.2% and 83.6% for recognition of entailment relation, contradiction relation and neural relation, respectively, which proves that this model has better overall performance than other advanced models. For the entailment relation between sentences, current solutions depend on the knowledge in the model's own learner corpus, which may result in misjudgment of some commonsense entailment problem. With in-depth research on the application of deep learning in the field of mapping knowledge domain, the next step is to explore how to utilize the mapping knowledge domain to integrate the learning of commonsense and semantic repository into the natural language inference model.

Acknowledgements

This research was financially supported by the National Natural Science Foundation of China (61472453, U1401256, U1501252, U1611264), Natural Science Research Projects in Colleges and Universities of Anhui Province (KJ2018A0780,KJ2020A1084) an Excellent Young Talents Support Projects of Anhui Province (gnfx2017177).

References

[1] P. Malakasiotis, I. Androutsopoulos, Learning textual entailment using SVMs and string similarity measures, in: Proc. ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, 2007.

[2] G. Dinu, R. Wang, Inference rules and their application to recognizing textual entailment, in: Proc. 12th Conference of the European Chapter of the Association for Computational Linguistics, 2009.

[3] P. Malakasiotis, I. Androutsopoulos, A generate and rank approach to sentence paraphrasing, in: Proc. Conference on Empirical Methods in Natural Language Processing, 2011.

[4] V. Jijkoun, M.D. Rijke, Recognizing textual entailment using lexical similarity, in: Proc. PASCAL Challenges Workshop on Recognising Textual Entailment, 2005.

[5] G.A. Miller, WordNet: a lexical database for English, Communications of the ACM 38(11)(1995) 39-41.

[6] T. Mikolov, Distributed representations of words and phrases and their compositionality, in: Proc. Neural Information Processing Systems, 2013.

[7] J. Pennington, R. Socher, C. Manning, Glove: global vectors for word representation, in: Proc. 2014 Conference on Empirical Methods in Natural Language Processing, 2014.

[8] C. Lyu, Y. Lu, D. Ji, B. Chen, Deep learning for textual entailment recognition, in: Proc. International Conference on TOOLS with Artificial Intelligence, 2016.

[9] S.R. Bowman, C. Potts, C.D. Manning, Recursive neural networks can learn logical semantics, in: Proc. 3rd Workshop on Continuous Vector Space Models and their Compositionality, 2015.

[10] S R. Bowman, G. Angeli, C. Potts, C.D. Manning, A large annotated corpus for learning natural language inference, in: Proc. 2015 Conference on Empirical Methods in Natural Language Processing, 2015.

[11] T. Rocktäschel, E. Grefenstette, K.M. Hermann, T. Kočiský, Reasoning about entailment with neural attention in: Proc. International Conference on Learning Representations, 2016.

[12] Y. Liu, C. Sun, L. Lin, X. Wang, Learning natural language inference using bidirectional LSTM model and inner-attention. <https://arxiv.org/abs/1605.09090>, 2016.

[13] W. Yin, H. Schütze, B. Xiang, B. Zhou, ABCNN: attention-based convolutional neural network for modeling sentence pairs, Transactions of the Association of Computational Linguistics 4(1)(2016) 259-272.

[14] S. Wang, J. Jiang, Learning natural language inference with LSTM, in: Proc. Human Language Technologies: The 2016 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2016.

[15] A.P. Parikh, O. Tckstrm, D. Das, J. Uszkoreit. A decomposable attention model for natural language inference, in: Proc. 2016 Conference on Empirical Methods in Natural Language Processing, 2016.

[16] S. Wang, J. Jiang. A compare-aggregate model for matching text sequences. <https://arxiv.org/abs/1611.01747>, 2016.

[17] Z .Wang, W. Hamza, R. Florian, Bilateral multi-perspective matching for natural language sentences in: Proc. 26th International Joint Conference on Artificial Intelligence, 2017.

[18] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, Enhanced LSTM for natural language inference, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics 1(1)(2017) 1657-1668.

[19] A. Williams, N. Nangia, S.R. Bowman, A broad-coverage challenge corpus for sentence understanding through inference. <https://arxiv.org/abs/1704.05426>, 2017.

[20] L. Mou, H. Peng, G. Li, Y. Xu, L. Zhang, Z. Jin, Discriminative neural sentence modeling by tree-based convolution, in: Proc. 2015 Conference on Empirical Methods in Natural Language Processing, 2015.

[21] J. Cheng, L. Dong, M. Lapata, Long short-term memory-networks for machine reading, in: Proc. 2016 Conference on Empirical Methods in Natural Language Processing, 2016.

[22] S.R. Bowman, C. Potts, C.D. Manning, Recursive neural networks can learn logical semantics, in: Proc. 3rd Workshop on Continuous Vector Space Models and their Compositionality, 2015.