Applied Intelligence (2021) 51:4305–4329
https://doi.org/10.1007/s10489-020-02156-7

Deep bi-directional interaction network for sentence matching

Mingtong Liu[1] · Yujie Zhang[1] · Jinan Xu[1] · Yufeng Chen[1]

Accepted: 16 December 2020 / Published online: 2 January 2021

Abstract

The goal of sentence matching is to determine the semantic relation between two sentences, which is the basis of many downstream tasks in natural language processing, such as question answering and information retrieval. Recent studies using attention mechanism to align the elements of two sentences have shown promising results in capturing semantic similarity/relevance. Most existing methods mainly focus on the design of multi-layer attention network, however, some critical issues have not been dealt with well: 1) the higher attention layer is easily affected by error propagation because it relies on the alignment results of preceding attentions; 2) models have the risk of losing low-layer semantic features with the increase of network depth; and 3) the approach of capturing global matching information brings about large computing complexity for model training. To this end, we propose a *Deep Bi-Directional Interaction Network* (DBDIN) to solve these issues, which captures semantic relatedness from two directions and each direction employs multiple attention-based interaction units. To be specific, the attention of each interaction unit will repeatedly focus on the original sentence representation of another one for semantic alignment, which alleviates the error propagation problem by attending to a fixed semantic representation. Then we design deep fusion to aggregate and propagate attention information from low layers to high layers, which effectively retains low-layer semantic features for subsequential interactions. Moreover, we introduce a self-attention mechanism at last to enhance global matching information with smaller model complexity. We conduct experiments on natural language inference and paraphrase identification tasks with three benchmark datasets SNLI, SciTail and Quora. Experimental results demonstrate that our proposed method can achieve significant improvements over baseline systems without using any external knowledge. Additionally, we conduct interpretable study to disclose how our deep interaction network with attention can benefit sentence matching, which provides a reference for future model design. Ablation studies and visualization analyses further verify that our model can better capture interactive information between two sentences, and the proposed components are indeed able to help modeling semantic relation more precisely.

Keywords Sentence matching · Deep interaction network · Deep fusion · Attention mechanism · Multi-layer neural network · Interpretability study

✉ Yujie Zhang
yjzhang@bjtu.edu.cn

Mingtong Liu
16112075@bjtu.edu.cn

Jinan Xu
jaxu@bjtu.edu.cn

Yufeng Chen
chenyf@bjtu.edu.cn

[1] School of Computer and Information Technology, Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing, China

1 Introduction

Sentence matching is a key technique in natural language processing (NLP), in which a system is asked to classify the logical and semantic relationship between two sentences [1]. This technique is widely applied to be the essential basis of many downstream NLP tasks that require modelling the relevance/similarity of two sentences. In natural language inference (NLI), sentence matching is utilized to judge whether a hypothesis sentence can reasonably be inferred from a premise sentence [2, 3]. In paraphrase identification (PI), it is utilized to identify whether two sentences express the equivalent meaning or not [4, 5], as shown in

Table 1 Sentence matching examples from natural language inference and paraphrase identification

Natural Language Inference		
Premise	A senior is waiting at the window of a restaurant Relationship that serves sandwiches.	Relation
Hypothesis	A person waits to be served his food.	Entailment
	A man is looking to order a grilled cheese sandwich.	Neutral
	A man is waiting in line for the bus.	Contradiction
Paraphrase Identification		
Sentence 1	She struck a deal with RH to pen a book today.	Relation
Sentence 2	She signed a contract with RH to write a book.	Paraphrase
	She denied today that she struck a deal with RH.	Non-Parpahrase

Table 1. It also has broad applications, e.g., information retrieval [6–8], summarization [9], question answering [10] and dialog system [10, 11]. Owing to its practical significance, sentence matching has attracted wide spread attention in NLP. However identifying logical and semantic relationship between two sentences is not trivial due to the problem of the semantic gap [12, 13]. The core issue for sentence matching is how to accurately model the related semantics between two sentences [1, 2, 14–16].

Recently, research done on sentence matching with deep neural networks [1, 2, 14, 17, 18] has accomplished a considerable superiority over traditional methods because of the better automatic features extraction. In the neural network-based methods, a matching model can be built in two types of methods. The first method is sentence-encoding based one [2, 19, 20], in which each sentence is separately encoded using RNN or CNN to a fixed-sized vector in a completely isolated manner. Then, a matching decision is made based on the two sentence vectors. Such separated sentence representation is unable to capture fine-grained (e.g., word and phrase level) relevance between two sentences, because two sentences have no interaction during the encoding procedure. Afterwards, sentence interaction method is proposed to model related semantic information between two sentences [1, 14–16, 21, 22], which obtains the representation of one sentence by depending on the representation of another sentence. This method allows the model to utilize interactive features between two sentences, e.g., attentive information, to learn sentence representation for the final decision. Specially, sentence interaction with multi-layer neural networks [1, 14, 21, 22] has shown improved performance to model semantic relatedness, in which multiple stacked attention layers are usually employed to model sentence interaction [14].

Through the above analyses, we can conclude that effectively exploiting both interactive features and deep network is very important for sentence matching. Despite the recent success of multi-layer interaction method, some critical issues still limit further performance improvements in deep sentence matching model. Firstly, higher attention layer is easily affected by error propagation, because the input of each attention relies on the alignment results learned in preceding attention layers [14]. When model captures incorrect alignments in the preceding attention layers, the attentive representation will affect the subsequential interactions. Meanwhile, although the related information from one sentence to another may be of different importance from that of the reversed direction [1], the same attentive weights are used by two directions [14]. Secondly, simple stacked attention layers can not effectively propagate semantic features learned at low layers to high layers, which makes the interactive learning is insufficient in multi-layer neural network because of the vanishing gradient problem [23, 24]. Thirdly, each interaction layer uses self-attention mechanism for capturing global information [14], and thus it brings about large computing complexity to the model training.

In this work, to tackle these problems, we propose a *Deep Bi-Directional Interaction Network* (DBDIN), an end-to-end neural network for sentence matching, which adopts a deep interaction method to enable the model capturing interactive features for performance improvement. We model semantic relatedness from two directions and employ multiple attention-based interaction units in each direction. To alleviate error propagation, the attention of each interaction unit is designed to attend to the original sentence representation of another one instead of interactive representation. Multiple interaction units allow one sentence to repeatedly read the information of another one, and therefore to better capture interactive features. Meanwhile, each direction specifically focuses on the other sentence in a directed way, which is able to learn different attentive weights to capture the direction-dependent relatedness. In this way, related semantic information at the word level can be well distinguished from different interaction directions, and thus these word-level finer-grained semantic relations will be effectively exploited for sentence matching. With the increment of interaction, the representation of one sentence can gradually encode the related semantics with the attended information from another sentence.

To better combine the advantages of attention and deep neural network for learning interactive features, we further introduce deep fusion mechanism, from which the semantic features learned at low layers can be selectively propagated to high layers for subsequential interactions, and it also makes better integration of low-level and high-level features to improve the overall performance of the model. By doing so, it alleviates the vanishing gradient problem for model training [1, 14, 21, 22], and therefore enabling our model to effectively learn deep interaction. Moreover, we introduce one layer of self-attention network after the cross sentence interaction to capture global matching information, in which the model complexity is greatly decreased compared to previous model using self-attention in each layer [14]. The advantage of self-attention is to capture long-distance semantic dependencies within each sentence, thus it can enhance global matching information for the final decision. Additionally, we conduct interpretable study to disclose how our deep interaction network with attention can benefit sentence matching, which provides a reference for future model design.

Overall, the main contributions of our work include the following aspects:

1. We propose a *Deep Bi-Directional Interaction Network* (DBDIN) that employs multiple attention-based interaction units for better modelling semantic relatedness between two sentences. Specifically, we make the attention at each interaction unit focusing on the original sentence representation of another one, which alleviates error propagation in multi-layer attention model and also enables model to capture direction-dependent relatedness. We further introduce deep fusion to aggregate and propagate low-layer semantic features for deep interaction, and self-attention mechanism to enhance global matching information. These proposed components are easily integrated into existing models.
2. Experimental results on the SNLI and SciTail datasets for natural language inference, and the Quora dataset for paraphrase identification demonstrate that the proposed model significantly improves accuracy over baselines without using any external knowledge.
3. We further conduct extensive ablation studies on the proposed several components, and perform visualization analyses to the learned attentions and sentence representations. These analyses explore intuitive interpretability of why our deep interaction network improves sentence matching, and provide a reference for future model design. These results further verify that our proposed model has the ability to capture more accurate semantic alignment of two sentences and can better integrate the learned semantic features of different interaction layers to improve the final decision.

The remainder of this paper is organized as follows. We introduce the related work and highlight the differences between work we did in this paper and previous studies in Section 2. In Section 3, we give a brief overview of our sentence matching framework. Section 4 elaborates the details of the proposed model. Section 5 describes the learning details of our model. Section 6 conducts experiments to verify the effectiveness of the proposed model. Section 7 presents in-depth analyses and discussion for matching results. Finally, we conclude this work and provide future direction in Section 8.

2 Related work

Sentence matching has been studied for many years. Early approaches focus on designing hand-craft features to capture n-gram overlapping, word reordering and syntactic alignments phenomena [25, 26]. This kind of method can work well on a specific task or dataset, but it's hard to generalize well to other tasks [1]. Recently, with the availability of large-scale annotated datasets such as SNLI [2], deep learning is rising a substantial interest in sentence semantic matching and has achieved some great progresses [2, 14, 20, 21, 27–30]. According to their learning ways, previous models can be classified into three categories.

2.1 Sentence-encoding based method

Some early neural network-based methods focus on designing encoder architecture, such as LSTM-based models [2, 20], CNN-based models [28], and Tree-LSTM-based models [31, 32], in which different neural architectures have their own advantages in learning semantic representation, such as LSTM for long-term dependency, CNN for local feature extraction and Tree-LSTM for structural information. As shown in Fig. 1, these models first separately encode each sentence as a vector representation with a neural network (e.g., LSTM). Then a neural network classifier is applied to predict their semantic relationship based on the two sentence representations. In this paradigm, two sentences have no interaction until arriving final phase. The advantage of this framework is that sharing parameters makes the model smaller and easier to train, and the learned sentence representations can be used for many other purposes [1]. However, this kind of framework ignores the explicit interaction between two sentences during the encoding procedure, and the sentence representation does not encode the related semantics from another sentence. It has been found that such separated sentence representation is often not sufficient to capture all the important information for deciding the final semantic relation [16, 33].

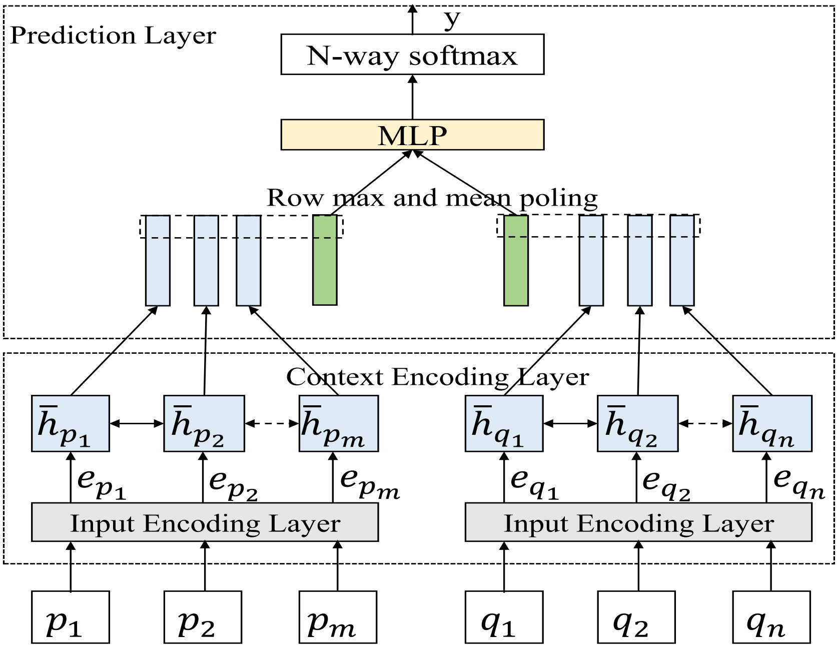


Fig. 1 An illustration of sentence matching models based on sentence-encoding method. This method focuses on learning vector representation of individual sentence and then predicts the semantic relationship between two sentences based on the two sentence vectors

2.2 Attention-based interaction method

Most recent works [14, 21, 30] focus on modelling interactive features between two sentences and often show better performance. These methods employ attention mechanism to align the elements of two sentences to model the semantic relatedness between them, which obtains the representation of one sentence by depending on the representation of another sentence. The attention-based framework decomposes sentence-level matching to lower-level matching. They build the interaction at different granularity (word, phrase and sentence level).

Under this framework, small semantic units of two sentences are first matched, then the matching results are aggregated by another network to make the final decision. One kind of methods is to model the conditional encoding, in which the encoding of one sentence can be affected by another sentence. Rockt$\ddot{a}$schel et al. [16] and Wang et al. [34] use LSTM and attention mechanism to read two sentences to produce a final representation, which can be regarded as interaction of two sentences. Another kind of method is to compute similarities between all the words or phrases of the two sentences to model multiple-granularity interactions of two sentences. Parikh et al. [15] propose a neural attention-based model that directly compares the relevant sub-components between two sentences. Furthermore, Wang et al. [1] and Chen et al. [21] propose a bidirectional matching framework with word-by-word interaction to model the semantic relatedness between two sentences. To improve the attention-based framework, Duan et al. [14] propose using multi-layer neural network with attention mechanism and show that multiple stacked attention layers can better extract interactive features to improve matching performance. Yang et al. [35] adopt augmented residual connections to consider more the lower-layer features for alignment. Besides the attention between two sentences, the self-attention mechanism is proposed to solve the limitations of RNN model on the long-term dependency problem for sentence matching [14], which aims to align the sentence with itself and has been used in a variety of tasks [36, 37].

Similar to previous work, we also adopt attention mechanism for modelling sentence matching. However, the approach taken by ours is different from them in at least four aspects. Firstly, in previous work, the attention is performed between two interactive representations. Different from previous approaches, we make the attention of each interaction unit takes the original sentence representation of another one as input to learn interactive features. Secondly, we model semantic relatedness from two directions to specially capture the direction-dependent relatedness, and employ multiple attention-based interaction units for each direction. Thirdly, we design deep fusion to better aggregate and propagate low-layer interactive features for subsequential interactions. Fourth, we introduce one layer of self-attention after cross sentence interaction to enhance global matching information instead of using self-attention at each layer [14]. Finally, our model effectively combines the advantages of attention mechanism and deep neural network, achieving a stronger ability of extracting across sentence semantic features to improve sentence matching performance. Our methods can be also combined to other strong systems, such as RE2 [35], to further improve sentence matching performance, and we leave it to the further work.

2.3 External knowledge based method

Although there are relatively large annotated data, it is still challenging for machines to learn all knowledge needed to perform complicated sentence matching from these annotated data. Previous work [7, 27, 29, 38–40] has shown that neural network-based representation learning models can benefit from leveraging external knowledge to achieve further performance improvement. These methods can be classified to two categories: explicit knowledge and implicit knowledge. Chen et al. [38] enrich neural network-based models with explicit knowledge (WordNet [41]), such as synonymy, antonyms, hypernymy, hyponymy, and co-hyponyms, to improve natural language inference. They consider external lexical-level semantic relation between two words collected in WordNet and use the inference knowledge to improve the attention-based word alignments between two sentences, achieving better performance. The second method uses implicit knowledge learned from a large unlabeled corpus, well known as pre-training model, such as ELMO [29] and BERT [27]. This method learns deep contextualized word representations with a language model, by which the knowledge is implicitly entailed in the word representations. Then the pre-trained model is fine-tuned with a specific data for applications, which has shown improved performance in sentence matching task.

However, these pre-trained models have especially large model parameters (such as 340M parameters in BERT) to learn, 80 times the general matching models (such as 4.3M parameters in ESIM [21]). Large number of model parameters will bring about large computing complexity and requires a lot of computing resources, which restricts model applications in case of insufficient computing resources.

In this work, we do not use any such external knowledge. Our work belongs to the attention-based interaction approaches with less model parameters (7.8M parameters) to learn, which is in line with the recent studies without using any external knowledge [14, 21, 30]. We mainly focus on model architecture that is more effective to capture related semantic information between two sentences, and we will explore the method of integrating external knowledge in the future work.

3 Overview of our sentence matching framework

In this section, we give a brief overview of our sentence matching framework, as shown in Fig. 2. Formally, we can define the sentence matching task as follows. Given two sentences $P = [p_1, \cdots, p_i, \cdots, p_m]$ and $Q = [q_1, \cdots, q_j, \cdots, q_n]$, the goal is to predict a label $y^* \in \mathcal{Y}$, where $\mathcal{Y}$ = {Entailment, Contradiction, Neutral} in natural language inference task and $\mathcal{Y} = \{0,1\}$ in paraphrase identification task, indicating the logical semantic relation between P and Q [1].

$$y^* = \arg\max_{y \in \mathcal{Y}} P_r(y|P, Q) \tag{1}$$

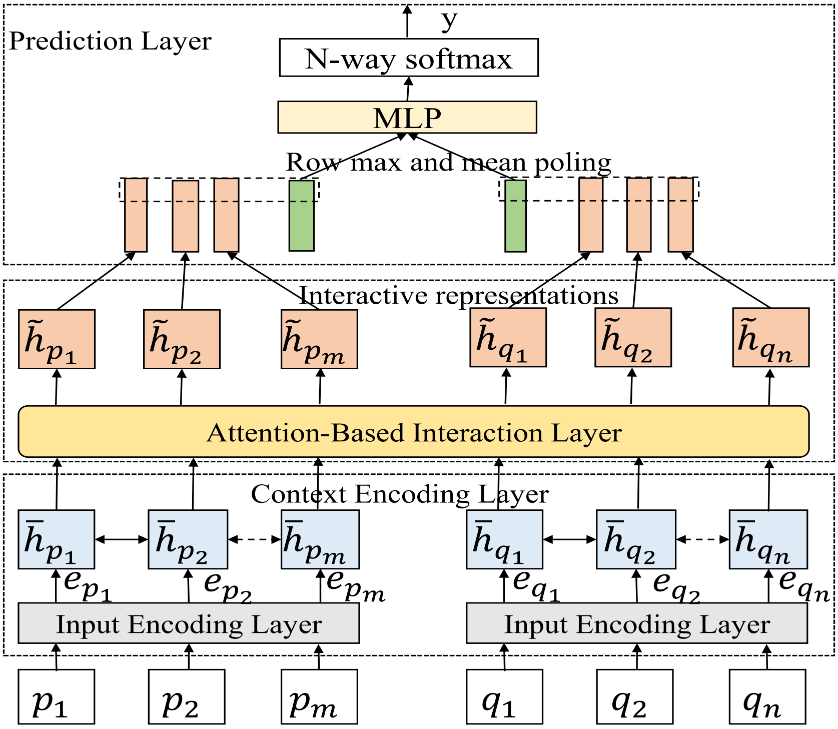


Fig. 2 An overview architecture of our sentence matching framework that employs attention mechanism to learn interactive representations for sentence semantic matching. The more details of our attention-based interaction layer are given in Section 4 and Fig. 3

The core element for neural network-based sentence matching models is to learn interactive sentence representation [1, 14–16]. Generally, the architecture of neural sentence matching mainly includes the following four components [1, 14]:

- **Input Encoding Layer** converts words to vector representations as input, where pre-trained word embeddings are usually used, e.g., GloVe [42].
- **Context Encoding Layer** incorporates context and sequence order into modeling for better word vector representations. This layer often uses CNN [28], LSTM [1, 14] and Tree-LSTM [21].
- **Attention-Based Interaction Layer** calculates word pair interactions using the outputs of the encoding layer to learn interactive sentence representations.
- **Prediction Layer** applies multilayer perceptron (MLP) and softmax function to predict the semantic relation according to the learned interactive sentence representations.

In this paper, we mainly focus on the attention-based interaction layer that has been proved to the most important part for improving sentence matching performance [1, 14, 35, 38]. We combine the advantages of attention mechanism and deep neural network, and propose a deep bi-directional interaction network to better model the related semantic information of two sentences. Figure 2 shows an overview architecture of our proposed model. In this section, we give an overall description of our model architecture. The details of our attention-based interaction layer are described in Section 4.

3.1 Input encoding layer

For the given sentence pairs $P = [p_1, \cdots, p_i, \cdots, p_m]$ with length m and $Q = [q_1, \cdots, q_j, \cdots, q_n]$ with length n, where p_i and q_j indicate the i-th and j-th word in P and Q respectively, the input encoding layer first converts words of P and Q into vectors $E_P = [e_{p_1}, \cdots, e_{p_i}, \cdots, e_{p_m}]$ and $E_Q = [e_{q_1}, \cdots, e_{q_j}, \cdots, e_{q_n}]$ by looking up M respectively, where $M \in \mathbb{R}^{d \times |V|}$ is an embedding table and each column in M represents a word. d is the dimension of embeddings and $|V|$ is the size of vocabulary V.

3.2 Context encoding layer

In natural language sentence, the meaning of a word usually depends on its context, in which the model is required to understand both lexical and compositional semantics [43, 44]. In order to acquire contextual information, we utilize Recurrent Neural Network (RNN) to encode sentences. RNN is designed to process sequential inputs and has shown powerful ability in NLP tasks. The sequential RNN calculates a new hidden state conditioned on the previous states, by which the word representations can incorporate contextual information. In our model, we employ bidirectional Long Short-Term Memory (BiLSTM) network [45] to encode sentences. The BiLSTM processes an input with two separate hidden layers, whose outputs are then used to as contextual word representations as (3) and (2).

$$\overline{h}_{p_i} = \text{BiLSTM}(e_{p_i}, \overrightarrow{\overline{h}}_{p_{i-1}}, \overleftarrow{\overline{h}}_{p_{i+1}}) \tag{2}$$

$$\overline{h}_{q_j} = \text{BiLSTM}(e_{q_j}, \overrightarrow{\overline{h}}_{q_{j-1}}, \overleftarrow{\overline{h}}_{q_{j+1}}) \tag{3}$$

Then, the two sentences are converted to vector representations $\overline{H}_P = [\overline{h}_{p_1}, \cdots, \overline{h}_{p_i}, \cdots, \overline{h}_{p_m}]$ and $\overline{H}_Q = [\overline{h}_{q_1}, \cdots, \overline{h}_{q_j}, \cdots, \overline{h}_{q_n}]$. Hereafter, we call $\overline{H}_P$ and $\overline{H}_Q$ as the original sentence representations of sentences P and Q, respectively, both of which do not consider interactive information from another sentence. In this work, we will use the $\overline{H}_P$ and $\overline{H}_Q$ as the targets of cross sentence attention to learn interactive sentence representations.

3.3 Attention-based interaction layer

Attention has, in recent years, demonstrated to be an effective mechanism for a neural network to "focus" on salient features of the input. Given an input state, attention allows the model to dynamically learn weights to indicate the importance of different parts of the inputs. It has been particularly successful for tasks requiring modeling of complex semantic relation. In this work, we employ attention mechanism to associate the relevant parts between two sentences to learn interactive features. Here, we first describe the general attention computing function and then introduce our sentence interaction method with attention.

Attention function An attention function can be described as mapping a query (Q) and set of key-value (K-V) pairs to an output, where the Q, K, V and outputs are all vectors [27]. The output is computed by an attention-weighted sum of the V, where the weights assigned to V are computed by a score function that uses the Q to attend to the corresponding K. The attention computation will produce an aligned context vector from V to capture the relevant information from another sentence, and it can be formulated as:

$$\widetilde{Q}_{Q \rightarrow K} = \text{Attention}(Q, K, V) = \text{softmax}(\text{score}(Q, K))V \tag{4}$$

where $\widetilde{Q}_{Q \rightarrow K}$ represents that query Q attend to key K to extract relevant information from V. The score function computes the relatedness of two vectors Q and K. The final score is the normalized weights by softmax function and then used to encode the entire vectors of V into an aligned vector. Intuitively, the information of V is more probably selected if it is more related to Q.

Sentence Interaction with Attention In this work, we model semantic relatedness from two directions, and employ multiple attention-based interaction units for each direction to capture the direction-dependent relatedness. The attention of each interaction unit takes the original sentence representation of another sentence as input to learn interactive features, in which each direction will specifically focus on the relevant parts of another sentence. Concretely, for two interaction direction $P \rightarrow Q$ and $Q \rightarrow P$, the attention used to capture the relevant information from another one can be formulated as:

$$\widetilde{H}_{P\rightarrow Q} = \text{Attention}(H_P, \overline{H}_Q, \overline{H}_Q) \quad (5)$$

$$\widetilde{H}_{Q\rightarrow P} = \text{Attention}(H_Q, \overline{H}_P, \overline{H}_P) \quad (6)$$

where H_P is the query vectors, and $\overline{H}_Q$ is the keys and values for interaction direction $P \rightarrow Q$. H_Q is the query vectors, and $\overline{H}_P$ is the keys and values for $Q \rightarrow P$. $H_{\cdot}$ is from the preceding interaction unit and $\overline{H}_{\cdot}$ is the original sentence representation of another one.

For the sake of brevity, we give the concrete attention computing method for interaction direction $P \rightarrow Q$ as (7) and (8). Equation (7) computes the relatedness scores between two representations, and (8) computes the aligned context vectors from another sentence. For the opposite direction $Q \rightarrow P$, we have the same computing method. We employ biaffine attention function [46] to compute the relatedness score of two representations h_{p_i} and $\overline{h}_{q_j}$.

$$A_{ij} = \text{score}(h_{p_i}, \overline{h}_{q_j}) = {h_{p_i}}^T W\overline{h}_{q_j} + \langle U_p, h_{p_i}\rangle + \langle U_q, \overline{h}_{q_j}\rangle \quad (7)$$

where $A \in \mathbb{R}^{m\times n}$ is the score matrix, m is the length of P and n is the length of Q. $W \in \mathbb{R}^{h\times h}$, U_p, $U_q \in \mathbb{R}^h$ are learnable parameters, h is dimension of vector representation, and $\langle \cdot, \cdot \rangle$ denotes the inner production operation. The first item ${h_{p_i}}^T W\overline{h}_{q_j}$ directly measures the relatedness score of two representations of words p_i and q_j. The second and third items measure how probable a word is taken as a related word to others, by which the score depends not only on the combination of two words but also on the word itself.

Next, for each word p_i in sentence P, the relevant semantic information in another sentence Q is extracted as a context vector according to the score matrix A as (8).

$$\widetilde{h}_{p_i} = \text{context}(A, \overline{H}_q) = \sum_{j=1}^{n} \frac{exp(A_{ij})}{\sum_{k=1}^{n} exp(A_{ik})}\overline{h}_{q_j} \quad (8)$$

where $\widetilde{h}_{p_i}$ is an attention-weighted representation of $\overline{H}_Q$, and the larger attentive weight indicates the corresponding information $\overline{h}_{q_j}$ in Q is more relevant to word p_i in P.

As shown in Fig. 2, after once attention-based interaction, sentences P and Q can be represented as $\widetilde{H}_{P\rightarrow Q} = [\widetilde{h}_{p_1}, \cdots, \widetilde{h}_{p_i}, \cdots, \widetilde{h}_{p_m}]$ and $\widetilde{H}_{Q\rightarrow P} = [\widetilde{h}_{q_1}, \cdots, \widetilde{h}_{q_j}, \cdots, \widetilde{h}_{q_n}]$, respectively, each of which encodes the relevant semantic information (i.e., interactive features) from another sentence.

3.4 Prediction layer

We employ a multilayer perceptron (MLP) classifier to determine the semantic relation between two sentences according to the learned interactive sentence representations. In the MLP classifier, a fixed-length vector is needed as input. To achieve this goal, we perform mean pooling and max pooling operation to convert the final sentence representations H_P of P and H_Q of Q into a fixed-length vector. For the representation vectors, each dimension represents different semantic features, in which the mean pooling averages each representation to preserve all of the information, and the max pooling selects the highlighting features to capture the significant properties. The computation can be defined as:

$$H_{P_{mean}} = \frac{1}{m}\sum_{i=1}^{m} h_{p_i}, H_{P_{max}} = \max_{i=1}^{m} h_{p_i} \quad (9)$$

$$H_{Q_{mean}} = \frac{1}{n}\sum_{j=1}^{n} h_{q_j}, H_{Q_{max}} = \max_{j=1}^{n} h_{q_j} \quad (10)$$

After that, sentences P and Q are represented as vectors $[H_{P_{mean}}; H_{P_{max}}]$ and $[H_{Q_{mean}}; H_{Q_{max}}]$ respectively, which encode all the related semantic information between two sentences.

Finally, we concatenate them together to get a fixed-length vector H as Chen et al. [21] and Duan et al. [14]. Then we pass H into a MLP classifier to predict the probability $P_r(\cdot)$ of each label, and the (1) is reformulated as (11).

$$H = [H_{P_{mean}}; H_{P_{max}}; H_{Q_{mean}}; H_{Q_{max}}] \quad (11)$$

$$P_r(\cdot|P, Q) = P_r(\cdot|H) = \text{softmax}(W_2\text{ReLU}(W_1 H + b_1) + b_2) \quad (12)$$

where W_1, W_2, b_1, b_2 are learnable parameters. $P_r(\cdot|P, Q)$ is the predicted label distribution.

4 Deep bi-directional interaction network

In this section, we elaborate the proposed *Deep Bi-Directional Interaction Network* (DBDIN) that combines attention mechanism and deep neural network to extract interactive features for learning sentence representation. Following the attention-based matching framework [1, 14, 15], we regard the semantic relation between two sentences P and Q as the relation aggregation of each pair words p_i and q_j, where $p_i \in P$, $i \in \{1, \cdots, m\}$, and $q_j \in$

Q, $j \in \{1, \cdots, n\}$. The relation of each pair p_i and q_j is defined as word-level relatedness, and the phrase- and sentence-level relatedness can be represented by the word-level relatedness, based on the compositional nature of sentence semantics [43, 44].

Figure 3 shows the details of our model architecture, in which the input encoding layer and context encoding layer are elaborated in Fig. 2 and we do not display their details here. As shown in Fig. 3, the DBDIN mainly consists of the following components: (1) cross sentence attention with original sentence representation to capture the relevant information from another sentence; (2) deep fusion to aggregate and propagate the learned attention information from low interaction layers to high interaction layers; and (3) self-attention mechanism to enhance global matching information. (1) and (2) are combined to form one interaction unit, as shown in Fig. 3b, where deep fusion is added after cross sentence attention. As shown in Fig. 3a, we use T attention-based interaction units that attend to the original sentence representation of another one to extract interactive features. Finally, we introduce one layer of self-attention to enhance global matching information after T cross sentence interaction units.

4.1 Cross sentence attention with original sentence representation

We use cross sentence attention to learn interactive features. Previous multi-layer model [14] performs attention between two parallel layers, in which one sentence attends to the interactive representation from the preceding layer of another one. As a result, semantics to be paid attention are uncertain and unstable for interaction because semantics are changed at different layers. This makes the attention in high layers is easily affected by error propagation. Different from previous work, we perform attention with original sentence representation, in which each attention will repeatedly focus on the original sentence representation of another one instead of the interactive representation. The attention will specifically focus on another sentence to be matched, and therefore the relatedness captured from one sentence to another one is different from that of the reversed direction.

For the sake of brevity, we take the interaction direction $P \rightarrow Q$ as an example to describe the attention computation. In the t-th interaction unit, where $t = \{1, \cdots, T\}$, the inputs contain two sentence representations, one is the interactive representation H_P^{t-1} of sentence P learned

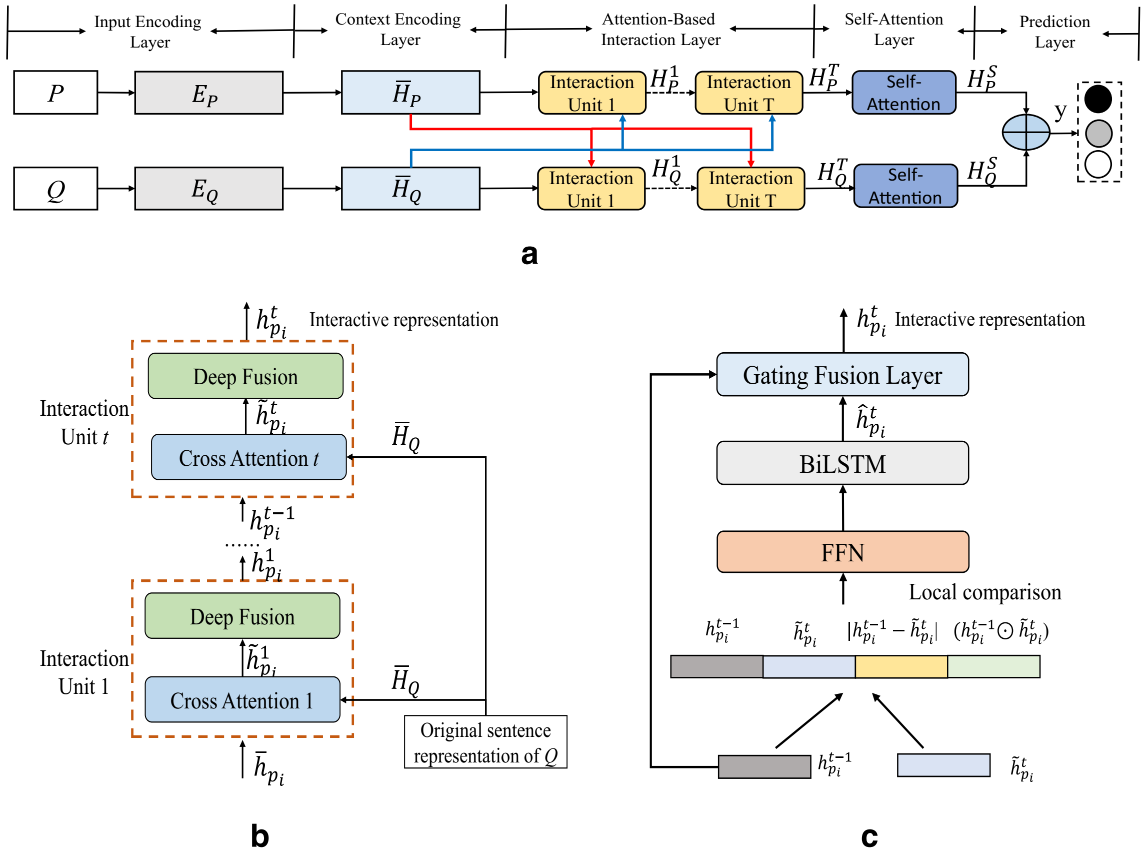


Fig. 3 An illustration of our *Deep Bi-Directional Interaction Network* (DBDIN). The interaction units in (**a**) are elaborated in (**b**), and the details of deep fusion are shown in (**c**)

in the preceding interaction units, and one is the original sentence representation $\overline{H}_Q$ of another one Q. When t=1, P also use its original sentence representation $\overline{H}_P$. The output is a new interactive representation $\widetilde{H}^t_P$ of P, which encodes the related semantics by aggregating the attended information from Q. For the opposite interaction direction $Q \rightarrow P$, we have the same computing method.

Concretely, given the attentive representation of P: $H^{t-1}_P = [h^{t-1}_{p_1}, \cdots, h^{t-1}_{p_i}, \cdots, h^{t-1}_{p_m}]$ and the original sentence representation of Q: $\overline{H}_Q = [\overline{h}_{q_1}, \cdots, \overline{h}_{q_j}, \cdots, \overline{h}_{q_n}]$, we first use (7) described in Section 3.3 to compute the unnormalized attentive weights A^t_{ij} for any pair $h^{t-1}_{p_i}$ and $\overline{h}_{q_j}$ between H^{t-1}_P and $\overline{H}_Q$. Next, we use the score matrix $A \in \mathbb{R}^{m \times n}$ to compute an attention-weighted representation $\widetilde{h}^t_{p_i}$ for each word p_i of P by using (8) described in Section 3.3. The attention vectors can be formulated as the following Eqs:

$$A^t_{ij} = \text{score}(h^{t-1}_{p_i}, \overline{h}_{q_j}) \tag{13}$$

$$\widetilde{h}^t_{p_i} = \text{context}(A^t, \overline{H}_Q) \tag{14}$$

After that, sentence P can be represented as $\widetilde{H}^t_P = [\widetilde{h}^t_{p_1}, \cdots, \widetilde{h}^t_{p_i}, \cdots, \widetilde{h}^t_{p_m}]$, where $\widetilde{h}^t_{p_i}$ is the interactive representation that encodes the relevant semantic information from sentence Q. Intuitively, the interaction can learn that the relatedness of some word pairs are more stronger than the relatedness between others according to different attentive weights. With the increment of interaction, the higher interaction layer can gradually capture the semantic relatedness at larger granularity, such as phrase-level relevance between two sentences.

4.2 Deep fusion

At each interaction unit, in addition to attention, we design deep fusion layer to aggregate the attention information through the network, by which the semantic features learned at low interaction layers can be effectively propagated to high layers for deep interaction, and it also makes better integration of low-level and high-level features, as shown in Fig. 3c. Here, we first describe local comparison operation and LSTM-based aggregation to fuse the attended information from another sentence, and then describe the gating deep fusion layer to aggregate the attention information learned at different interaction units.

Local Comparison Operation After extracting the relevant information from another sentence, a trivial next step would be to pass the concatenation of the $\widetilde{h}^t_{p_i}$ and $h^{t-1}_{p_i}$ to the following layer. In interaction operation, the concatenation can retain all the information [21, 30]. However, the model would suffer from the absence of similarity and relatedness information. Besides, for many sentence matching problems, we also note that it is helpful to check how similar or related at the word level for measuring the semantic similarity or relatedness of the two sentences. Therefore, we first perform a local comparison operation at the word level.

We consider the following comparison functions that measure the similarity and relatedness respectively [21, 30], which operates on two vectors in an element-wise manner. We calculate the element-wise substraction and element-wise multiplication between two vector representations $\widetilde{h}^t_{p_i}$ and $h^{t-1}_{p_i}$, where $h^{t-1}_{p_i}$ is the representation learned at the preceding layer and $\widetilde{h}^t_{p_i}$ is the attended representation from another sentence.

$$\text{Substraction}: \quad h^{sub}_{p_i} = f(h^{t-1}_{p_i}, \widetilde{h}^t_{p_i}) = (h^{t-1}_{p_i} - \widetilde{h}^t_{p_i}) \odot (h^{t-1}_{p_i} - \widetilde{h}^t_{p_i}) \tag{15}$$

$$\text{Multiplication}: \quad h^{mul}_{p_i} = f(h^{t-1}_{p_i}, \widetilde{h}^t_{p_i}) = h^{t-1}_{p_i} \odot \widetilde{h}^t_{p_i} \tag{16}$$

Note that the operator $\odot$ is element-wise multiplication. For both comparison functions, the resulting vectors $h^{sub}_{p_i}$ and $h^{mul}_{p_i}$ have the same dimensionality as $h^{t-1}_{p_i}$ and $\widetilde{h}^t_{p_i}$. We can see that the substraction is closely related to Euclidean distance that measures the similarity of two vectors, while the multiplication is closely related to cosine similarity that measures the relatedness of two vectors.

Then the element-wise substraction and multiplication are concatenated with the original vectors. We use a fully connected feed-forward network (FFN) with ReLU activations [47] to project the concatenated vectors from $4h$-dimensional vector space into a h-dimensional vector space, which operation helps the model to capture deeper interaction information and also reduces the complexity of vector representation.

$$h^c_{p_i} = [h^{t-1}_{p_i}; \widetilde{h}^t_{p_i}; h^{sub}_{p_i}; h^{mul}_{p_i}] \tag{17}$$

$$\widetilde{h}^t_{p_i} = \text{ReLU}(W^t_h h^c_{p_i} + b^t_h) \tag{18}$$

where $[\cdot; \cdot; \cdot; \cdot]$ refers to the concatenation operation, $W^t_h \in \mathbb{R}^{4h \times h}$, $b^t_h \in \mathbb{R}^h$ are learnable parameters.

Aggregation of Local Comparison Results The local comparison operation performs word-level information fusion. However, the understanding of complex semantic relatedness may rely on the contextual interaction information. Based on this consideration, we apply a recurrent BiLSTM network to further gather the sequential interaction vectors. The BiLSTM aggregation can be formulated as following.

$$\widehat{h}^t_{p_i} = \text{BiLSTM}(\widetilde{h}^t_{p_i}, \overrightarrow{h}^t_{p_{i-1}}, \overleftarrow{h}^t_{p_{i+1}}) \tag{19}$$

The BiLSTM inputs are the local comparison results. This aggregation is performed in a sequential manner to enhance local interaction vector with context interaction information that is important for measuring sentence-level semantic relatedness.

Gating Deep Fusion Layer Although sentence interaction has benefited from deep neural network [14] that takes the output of current attention layer as the input of next layer, it still suffers from some issues. First, only feeding the results of current attention layer into the next attention layer has the risk of losing low-layer semantic information [23, 24]. Second, if the current attention layer captures some alignment errors, the next layer will only have the incorrect information as input [48]. Besides, network training becomes more difficult with increasing depth because of the vanishing gradient problem [23, 24, 48, 49].

To solve the issues identified above with multi-layer attentions, we propose the gating deep fusion mechanism. Compared to the previous multi-layer attention model [14], we allow next interaction unit to utilize not only the attention information from current interaction unit but also from the previous units. This allows for the following benefits: 1) By fusing both lower-layer features and high-layer features, it should help to improve the performance of the attention; 2) By receiving the earlier interaction information, it provides the next attention with a second chance to revise the attention errors presented at the current layer; 3) Furthermore, it helps to mitigate the gradient vanishing problem for model training.

Inspired by previous work [24, 45, 49–51] that show adding paths with linear connection between two layers can effectively propagate information and train deep network, we design a gating deep fusion layer. The gating deep fusion layer is based on the interaction results at both the current and the preceding interaction units, which learns adaptively controlling how much the semantic information in the preceding layers to be propagated to the following layers.

First, we gather the representations from both the current interaction unit and the preceding interaction unit as following:

$$h^g_{p_i} = f(h^{t-1}_{p_i}, \widehat{h}^t_{p_i}) = [h^{t-1}_{p_i}; \widehat{h}^t_{p_i}; h^{t-1}_{p_i} \odot \widehat{h}^t_{p_i}] \tag{20}$$

where $\widehat{h}^t_{p_i}$ is from the current interaction unit, and $h^{t-1}_{p_i}$ is from the preceding interaction unit, respectively. $\odot$ is element-wise multiplication.

Then, based on the representation $h^g_{p_i}$, we design two gates $r^t_{p_i}$ and $z^t_{p_i}$ to control information propagation. The forget gate $r^t_{p_i}$ decides whether the previous semantic information is ignored. The update gate $z^t_{p_i}$ selects whether the learned semantic representation is to be updated with a new interactive representation $\widetilde{c}^t_{p_i}$. The detailed computations of $r^t_{p_i}$, $z^t_{p_i}$, $h^t_{p_i}$ and $\widetilde{c}^t_{p_i}$ are shown in (21)–(24).

$$r^t_{p_i} = \sigma(W^t_r h^g_{p_i} + b^t_r) \tag{21}$$

$$z^t_{p_i} = \sigma(W^t_z h^g_{p_i} + b^t_z) \tag{22}$$

$$\widetilde{c}^t_{p_i} = \tanh(W^t_c[r^t_{p_i} \odot h^{t-1}_{p_i}; \widehat{h}^t_{p_i}] + b^t_c) \tag{23}$$

$$h^t_{p_i} = z^t_{p_i} \odot h^{t-1}_{p_i} + (1 - z^t_{p_i}) \odot \widetilde{c}^t_{p_i} \tag{24}$$

where σ is a sigmoid function, the value of $r^t_{p_i}$ and $z^t_{p_i}$ is between 0 and 1. W^t_* and b^t_* are learnable parameters. Intuitively, the values of $r^t_{p_i}$ and $z^t_{p_i}$ closing to 1 imply that the more semantic information from the previous interaction units will be propagated to the following interaction units, while closing to 0 imply that the semantic information of previous interaction units is less propagated and the new interactive information is used to update the sentence semantic representation.

After the t-th cross sentence interaction unit, each word p_i in sentence P is newly represented as $h^t_{p_i}$ that captures the relevant information from another sentence Q, and learns new interactive representation for the final semantic relation judging between P and Q.

Similarly, we build multiple interaction units from the opposite direction $Q \rightarrow P$, implying that the sentence Q will focus on the relevant semantic information from the sentence P with attention mechanism. At each interaction unit, the sentence Q attends to sentence P, which will learn from the original sentence representation $\overline{H}_P$ of P to derive the interactive representation $h^t_{q_j}$ for each word q_j of Q.

4.3 Self-attention layer

After cross sentence interaction, we further introduce a self-attention mechanism to enhance global matching information, as shown in Fig. 3a. The self-attention directly computes semantic relatedness between two representations regardless of their distance. Previous studies [14, 36, 37] have shown that self-attention is specially helpful to capture long-distance context information for modeling sentence. Our motivation of using self-attention for sentence matching is to capture long-distance interaction information within each sentence to enhance global matching.

Concretely, for sentence P, given its interactive representation $H^T_P = [h^T_{p_1}, \cdots, h^T_{p_i}, \cdots, h^T_{p_m}]$, computed after T cross sentence interaction units. We first compute a self-attentive score matrix $S^s \in \mathbb{R}^{m \times m}$ by using (7) described in Section 3.3:

$$S^s_{ij} = \text{score}(h^T_{p_i}, h^T_{p_j}) \tag{25}$$

where S^s_{ij} indicates the relatedness score between the two interactive representations $h^T_{p_i}$ and $h^T_{p_j}$ of the i-th and j-th word in P.

Then, we use the self-attentive score matrix $S^s \in \mathbb{R}^{m\times m}$ to compute a global context vector $\widetilde{h}^s_{p_i}$ for each word in P by using (8) described in Section 3.3.

$$\widetilde{h}^s_{p_i} = \text{context}(S^s, H^T_P) \tag{26}$$

Intuitively, the $\widetilde{h}^s_{p_i}$ can capture all contextual interaction information within sentence P, and therefore enhancing global sentence-level matching results.

After that, we also perform a comparison function and BiLSTM fusion, as described in Section 4.2, to better aggregate the global matching information, as (27)–(29).

$$\overline{h}^s_{p_i} = [h^T_{p_i}; \widetilde{h}^s_{p_i}; \mid h^T_{p_i} - \widetilde{h}^s_{p_i} \mid; h^T_{p_i} \odot \widetilde{h}^s_{p_i}] \tag{27}$$

$$\widetilde{h}^s_{p_i} = \text{ReLU}(W^s_h \overline{h}^s_{p_i} + b^s_h) \tag{28}$$

$$\widehat{h}^s_{p_i} = \text{BiLSTM}(\widetilde{h}^s_{p_i}, \overrightarrow{\widehat{h}}^s_{p_{i-1}}, \overleftarrow{\widehat{h}}^s_{p_{i+1}}) \tag{29}$$

Similarly, we conduct self-attention to sentence Q to derive the semantic representation $\widehat{h}^s_{q_j}$ for each word q_j of Q.

Finally, to rich the interactive representation, we further fuse the above semantic representations learned by cross sentence interaction and self-attention to get the final representations for two interaction directions $P \to Q$ and $Q \to P$. The computation is as follows:

$$h^r_{p_i} = h^T_{p_i} + \widehat{h}^s_{p_i} \tag{30}$$

$$h^r_{q_j} = h^T_{q_j} + \widehat{h}^s_{q_j} \tag{31}$$

where the semantic representations $h^r_{p_i}$ and $h^r_{q_j}$ are constructed from cross sentence interaction units and then are enhanced by global matching information with self-attention.

Then, the two sentences P and Q are converted to representations $H^r_P = [h^r_{p_1}, \cdots, h^r_{p_i}, \cdots, h^r_{p_m}]$, and $H^r_Q = [h^r_{q_1}, \cdots, h^r_{q_j}, \cdots, h^r_{q_m}]$, which encodes the related semantic information between them. Finally, H^r_P and H^r_Q are passed into the prediction layer as input to predict the semantic relation between the two sentences.

5 Model learning

In this section, we will introduce the details about the model learning, which can be classified into three parts: model input, loss function and model configuration.

5.1 Model input

In order to represent each input word better, inspired by previous work [12, 52], we concatenated three types of vectors: a pre-trained vector $e^{pre}_i \in \mathbb{R}^{d_1}$, a learnable vector $e^{learn}_i \in \mathbb{R}^{d_2}$ for each word type, and a learnable vector $e^{pos}_i \in \mathbb{R}^{d_3}$ for the POS tag of the word. The pre-trained word vector includes rich semantic information learned from a large scale of unlabeled corpus, the learnable word vector can learn task-specific word representation, and the POS tag further riches word representation. We used NLTK[1] to acquire POS tags. We applied a nonlinear transformation ReLU [47] to the concatenated vector to get the final word embedding $e_i \in \mathbb{R}^d$.

$$e_i = \text{ReLU}(W_e[e^{pre}_i; e^{learn}_i; e^{pos}_i] + b_e) \tag{32}$$

where $W_e \in \mathbb{R}^{(d_1+d_2+d_3)\times d}$ and $b_e \in \mathbb{R}^d$ are a weight matrix and a bias vector, respectively.

5.2 Loss function

We employed cross-entropy as the loss function since the goal is to make the correct classification. Considering the model complexity, we also added $l2$-norm of all learnable parameters to the final loss function. The following is the loss function for the output of classifier, which can be formulated as:

$$\mathscr{J}(\Theta) = -\frac{1}{N}\sum_{i=1}^{N} \log P_r(y^{(i)}|P^{(i)}, Q^{(i)}; \Theta) + \frac{1}{2}\lambda\|\Theta\|^2_2 \tag{33}$$

where $(P^{(i)}, Q^{(i)})$ are the sentence pairs, and $y^{(i)}$ denotes the corresponding annotated label for the i-th instance, N is the number of instances in the training set. λ is a regularization weight controlling the complexity and Θ denotes all the learnable parameters of our model. Our objective adding these two terms is differentiable, allowing the model to be efficiently trained with gradient descent algorithm in an end-to-end way.

5.3 Model configuration

In order to get the best performance, we have tuned the hyper-parameters on the development set. We used the parameters that perform the best on the development set to evaluate the model performance on the test set. The values of the hyper-parameters are illustrated as follows:

For input encoding layer, we used the pre-trained word vectors (*Glove* 840B) [42], in which the dimension was set as 300. We set the learnable word vectors and POS vectors to 30 dimensions. The final projected word embedding was set as 300 dimensions. In order to reduce learning complexity, we did not update the pre-trained word vectors during training. For BiLSTM layer, all of the hidden states were set as 300 dimensions. The ReLU layers for each comparison operation were set as 300 dimensions. For the final classifier, we used two-layer MLP with 1024-dimensions hidden states. For all datasets, we used 3 cross

[1] http://www.nltk.org/

Table 2 Statistics of datasets: SNLI, SciTail and Quora. Avg.L refers to average length of a pair of sentences

	Train	Dev	Test	Avg.L		Vocab
SNLI	549K	9.8K	9.8K	14	8	36K
SciTail	23K	1.3K	2.1K	17	12	24K
Quora	384K	10K	10K	12	12	107K

sentence interaction units and 1 layer of self-attention. The parameters were shared for two interaction directions in the t-th layer, and different layers had different parameters.

For model learning, the batch size was set as 64 for SNLI and Quora, 32 for SciTail, because the first two datasets has more training samples, and a larger batch size will speed up model training. We used the Adam method with $\beta_1 = 0.9$, $\beta_2 = 0.999$ [53] for model optimization. We set the initial learning rate to 5e-4 with a decay ratio of 0.95 for each epoch, and $l2$ regularizer strength to 6e-5. To effectively train model, we performed batch normalization regularization [54] to the pre-trained word vectors and projected word embeddings for each training mini-batch. To prevent over-fitting, we used dropout regularization [55] with a drop rate of 0.2. Specially, dropout was applied after batch normalization.

For initialization, we randomly set all the learnable parameters with the uniform distribution in the range between [-0.01,0.01]. We implemented our model using open source deep learning platform pytorch[2]. The models were trained on 1 NVIDIA GTX1080 GPU card.

6 Experiments

In this section, we conducted experiments to evaluate the effectiveness of our proposed model on two sentence matching tasks with three benchmark datasets: (1) SNLI and SciTail datasets for natural language inference; (2) Quora dataset for paraphrase identification.

6.1 Dataset description

SNLI is a natural language inference dataset proposed by Bowan et al. [2]. This dataset contains 570,152 human-written English sentence pairs, each labeled with one of the following relations: $\mathcal{Y}$ = {Entailment, Contradiction, Neutral}, where entailment indicates Q can be inferred from P, contradiction indicates Q cannot be true condition on P, and neutral means P and Q are irrelevant to each other. We followed the same data split as in Bowan et al. [2].

SciTail is an entailment classification task similar to SNLI dataset, but the semantic relation in SciTail is binary, where $\mathcal{Y}$ = {Entailment, Neutral}. Different SNLI, this dataset is created from natural sentences rather than written under the constraint of predefined rules and the language skills of humans. This dataset contains 23k pairs for training, 1,304 pairs for development and 2,126 pairs for testing [56]. Notably, the premise and the corresponding hypothesis have high lexical similarity for both the entailed and the non-entailed (neutral) pairs, which makes the task particularly difficult made evident by the low accuracy. We followed the same data split as in Khot et al. [56].

Quora is a paraphrase identification task. This dataset consists of over 400,000 question pairs and $\mathcal{Y} = \{0, 1\}$, where $y = 1$ means that P and Q are paraphrase of each other, and $y = 0$ means they are not paraphrases. We followed the same data split as in Wang et al. [1].

The detailed statistical information of the three datasets is shown in Table 2.

6.2 Ensemble strategy

The ensemble strategy has been proved to be effective in improving model accuracy for sentence matching [1, 12, 14]. The training mechanism of neural network is based on stochastic gradient descent, therefore different initialization of network parameters will lead to different training results. The ensemble models use multiple learning results from different initialized networks, which improves the prediction accuracy of the final task by alleviating network randomness. Following Duan et al. [14], our ensemble model averages the probability distributions from three individual single models to decide the final result, and each of them has the same architecture but different parameter initialization.

6.3 Baselines

We compared our model with several state-of-the-art baseline models in the sentence matching field. We mainly compared our model with previous sentence-encoding methods and attention-based interaction methods.

The sentence-encoding based methods:

- LSTM encoder [2] is a LSTM-based model that uses LSTM network to encode the premise and the hypothesis respectively.

[2] https://pytorch.org/

- tree-based CNN encoder [28] uses CNN network to encode sentences.
- SPINN [32] integrates tree-structured LSTM to encode sentence with syntactic information.
- DRCN [57] adopts densely-connected network to better generate sentence representation.
- SAN [36] utilizes the masked multi-head attention with distance to obtain sentence representations, which can effectively encode sentence semantics from multiple aspects.

The attention-based interaction methods:

- LSTM with attention [16] extends the general LSTM architecture with attention mechanism to read the information of another sentence.
- mLSTM [34] explicitly enforces word-by-word interaction between the hypothesis and the premise.
- LSTMN with deep attention fusion [58] exploits LSTM with memory which links the current word to previous words stored in memory with attention.
- re-read LSTM [59] uses a LSTM variant which considers the attention vector of another sentence as an inner state of LSTM.
- Decomposable attention model [15] decomposes sentence-level interaction to word-by-word interaction model with attention, and uses pre-trained word vector without relying on any word-order information.
- btree-LSTM [60] proposes an attention architecture with a complete binary tree-LSTM encoder (btree-LSTM).
- DIIN [12] hierarchically extracts semantic features from interaction space by using convolutional feature extractors.
- BiMPM [1] designs multiple parametric attention functions for interaction.

Table 3 Single model performance for natural language inference on SNLI dataset

Models	Params	Train	Test
Sentence encoding based method			
(1) LSTM encoder [2]	3.0M	99.7	78.2
(2) tree-based CNN encoder [28]	3.5M	83.3	82.1
(3) Tree-LSTM encoder (SPINN) [32]	3.7M	83.9	80.6
(4) Self-attention encoder (SAN) [36]	3.1M	89.6	86.3
(5) DRCN [57]	5.6M	–	86.5
Attention-based interaction method			
(6) LSTM with attention [16]	0.6M	85.3	83.5
(7) mLSTM [34]	1.9M	92.0	86.1
(8) LSTMN with deep attention fusion [58]	3.4M	88.5	86.3
(9) re-read LSTM [59]	2.0M	90.7	87.5
(10) Decomposable attention model [15]	0.6M	89.5	86.8
(11) btree-LSTM [60]	2.0M	88.6	87.6
(12) DIIN [12]	4.4M	91.2	88.0
(13) BiMPM [1]	1.6M	90.9	87.5
(14) ESIM [21]	4.3M	92.6	88.0
(15) DR-BiLSTM [22]	7.5M	94.1	88.5
(16) DRCN [57]	6.7M	93.1	88.9
(17) RE2 [35]	2.8M	94.0	88.9
(18) AF-DMN [14]	-	94.5	88.6
External knowledge based method			
(19) KIM [38]	4.3M	94.1	88.6
(20) ELMO [29]	8.0M	91.6	88.7
(21) DMAN [39]	9.2M	95.4	88.8
(22) SLRC (ELMO+SRL) [61]	–	–	89.1
(23) SLRC (BERT+SRL) [61]	308M	95.7	91.6
(24) SemBERT [62]	339M	94.4	91.9
This work			
(25) DBDIN	7.8M	93.5	88.8

- ESIM [21] incorporates the traditional sequential LSTM and tree LSTM for better semantic encoding and interaction.
- DR-BiLSTM [22] models interaction by processing the hypothesis conditioned on the premise results.
- RE2 [35] adopts augmented residual connections to consider more the lower-layer features for alignment.
- AF-DMN [14] proposes a multi-layer interaction network based on attention mechanism, and shows stacked cross attention and self-attention layers can better extract interactive features for sentence matching.

6.4 Experiments on natural language inference

Results on SNLI We verified the effectiveness of our model on SNLI dataset and compared our model with the following published models. The results are shown in Tables 3 and 4. These previous models can be categorized into three groups:

(1) The first group of models is based on sentence-encoding method. These models mainly focus on designing encoder architecture. We compared our model with LSTM-based model [2], tree-based CNN [28], SPINN [32], DRCN [57] and SAN [36]. Among these sentence-encoding based models, DRCN [57] and distance-based self-attention network (SAN) [36] are the current state-of-the-art models. These models separately encode each sentence as a vector representation in a completely isolated manner, and decide semantic relationship based on the two sentence representations. The advantage of this method is that less parameters make the model smaller and easier to train. However, the final sentence representation can not encode the fine-grained related semantics from another sentence, which often leads to the model to be insufficient for matching sentence pairs where complex reasoning is required.

(2) The second group of models is based on attention mechanism. These models obtain the representation of one sentence by depending on the representation of another sentence, which extracts attentive features to learn interactive sentence representation. These methods can be classified into two categories according to their interaction ways.

One kind of methods is to model the conditional encoding, in which the encoding of one sentence can be affected by another sentence. Previous models following this architecture include LSTM with attention [16], mLSTM [34], LSTMN with deep attention fusion [58], and re-read LSTM [59]. These methods focus on designing interactive encoder, which uses attention to read the information of another sentence during the procedure of encoding one sentence.

Another kind of methods is to compute similarities between all the words or phrases of two sentences to model multiple-granularity interactions. Previous models following this architecture include Decomposable attention model [15], btree-LSTM [60], DIIN [12], BiMPM [1], ESIM [21], DR-BiLSTM [22], DRCN [57], RE2 [35] and AF-DMN [14]. These interaction methods have achieved higher accuracy because of better modeling related semantics between two sentences. Among these models, multi-layer interaction network based on attention mechanism often obtains better performance [14, 35].

In Table 3, our single DBDIN model achieves 88.8% test accuracy in SNLI test set. For comparison with DRCN [57] and RE2 [35], our model obtains a bit low score. We analyzed that both DRCN and RE2 employ deeper networks that benefit sentence matching, such as 5 cross attentions in DRCN. We used 3 cross attentions and reported the results. We also verified the impact of network depth in Section 7.1.1 and shown that our model can be further improved with the increase of network depth. Moreover, we also reported the ensemble result in Table 4, and the test accuracy is 89.5%. The comparison results show that our model can effectively improve sentence matching performance on single and ensemble scenarios on SNLI dataset. As described in Section 4, DBDIN utilizes cross sentence attention with original sentence representation and deep fusion. DBDIN can pay close attention to another sentence at each step and the multiple interaction units allow

Table 4 Ensemble model performance for natural language inference on SNLI dataset

Models	Params	Train	Test
(1) DIIN (ensemble) [12]	17.0M	92.3	88.9
(2) BiMPM (ensemble) [1]	6.4M	93.2	88.8
(3) ESIM (ensemble) [21]	7.7M	93.5	88.6
(4) DR-BiLSTM (ensemble) [22]	45.0M	94.8	89.3
(5) AF-DMN (ensemble) [14]	-	94.9	89.0
(6) KIM (ensemble) [38]	43.0M	93.6	89.1
This work			
(7) DBDIN (ensemble)	23.4M	94.2	89.5

the model to better extract interactive features by repeatedly reading another sentence to be matched. The deep fusion can better aggregate and propagate the semantic features from low interaction units to high interaction units. The self-attention layer can effectively enhance global matching information. Therefore, the related semantics can be fully explored in an interactive way.

(3) The third group of models is based on external knowledge, such as WordNet [38], discourse marker prediction [39], semantic role labeling (SRL) [61], and pre-trained language model [27, 29, 62]. These models introduce other learning objectives or training data to obtain the representation of one sentence, intuitively, more learning signals and training data often can obtain improved performance. KIM [38] uses WordNet knowledge base [41] to enhance the learning of word-level semantic relation and obtains 0.6 improvement on the basis of ESIM [21]. They integrate the knowledge-based sore of word pairs into the cross sentence attention to better learn word alignment in term of word-level semantic relation, where knowledge about synonymy, antonymy, hypernymy and hyponymy between given words may help model alignment between premises and hypotheses; knowledge about hypernymy and hyponymy may help capture entailment; knowledge about antonymy and co-hyponyms (words sharing the same hypernym) may benefit the modeling of contradiction. DMAN [39] transfers knowledge from another supervised task, and use discourse marker "so" or "but" to help model learning the logical relationship between two sentences. ELMO [29], SLRC [61] and SemBERT [62] adopt pre-training language model technique using a large scale of unlabel corpus. Specially, SLRC [61] and SemBERT [62] show that integrating supervised semantic role labeling can further improve the quality of sentence representation.

Although ELMO [29], BERT [27] and SemBERT [62] have been well known as pre-trained language model for acquiring contextual word vectors to improve sentence matching, these models have large computing complexity (i.e., especially large model parameters and large training data). BERT and SemBERT have about 340M parameters to learn, and use the BooksCorpus (800M words) and English Wikipedia (2,500M words) as the pre-training corpus. The pre-training model needs not only large computing resources but also a long time, which restricts model application in case of insufficient computing resources. Our proposed model has less computing complexity (7.8M parameters) than BERT (340M parameters) and does not rely on any external knowledge, but obtains competitive performance. We will conduct the pre-training technique with our model in the future. In this paper, we presented a lightweight neural model, and mainly evaluated the contribution of our proposed neural architecture to sentence matching.

Results on SciTail We also verified the effectiveness of our model on SciTail dataset. In this dataset, the premise and the corresponding hypothesis have high lexical similarity for both the entailed and the non-entailed (neutral) pairs, which makes it particularly difficult for model to learn semantic features to effectively identify the semantic relation. Khot et al. [56] report that SciTail challenges typical attention-based models that show outstanding performance on SNLI, such as DecompAtt model [15] and ESIM model [21].

We compared our model with the following published models on SciTail dataset, and shown the results in Table 5. The first five models in Table 5 are all implemented in the work of Khot et al. [56]. DGEM proposed by Khot et al. [56] is a graph-based attention model for encoding sentence representation, and they show that syntactic structure information is helpful for understanding the semantic relation between two sentences. Yin et al. [63] propose deep explorations of inter-sentence interaction (DEISTE), and use attention mechanism to model the word-level relations between two sentences. CAFE [52] improves previous comparison function [30] by compressing alignment vectors into scalar valued features. Among these models, RE2 [35] is the current state-of-the-art model that considers more the lower-layer features for alignment. AF-DMN (re-imp) is our re-implementation of the model in Duan et al. [14] in which the original work do not report the results on this dataset.

On this dataset, our single DBDIN significantly outperforms previous models, achieving 86.8% accuracy on the SciTail test set. Compared to previous strong neural models AF-DMN [14] and RE2 [35] with multi-layer attentions, our proposed model shows better performance. Results on SciTail dataset further demonstrate that the proposed methods have the ability to better capture interactive features for matching sentence pairs that involve more complicated reasoning in natural language inference. Finally, our model achieves improved performance on the challenging SciTail dataset.

Table 5 Performance for natural language inference on SciTail dataset

Models	Dev	Test
(1) Majority class [56]	63.3	60.3
(2) Ngram [56]	65.0	70.6
(3) DecompAtt [15]	75.4	72.3
(4) ESIM [21]	70.5	70.6
(5) DGEM [56]	79.6	77.3
(6) DEISTE [63]	82.4	82.1
(7) CAFE [52]	–	83.3
(8) RE2 [35]	–	86.0
(9) AF-DMN (re-imp) [14]	87.2	84.4
(10) DBDIN	88.9	86.8

Table 6 Performance for paraphrase identification on the Quora dataset

Models	Test
(1) Siamese-CNN [1]	79.60
(2) Multi-Perspective-CNN [1]	81.38
(3) Siamese-LSTM [1]	82.58
(4) Multi-Perspective-LSTM [1]	83.21
(5) L.D.C [64]	85.55
(6) BiMPM [1]	88.17
(7) AF-DMN [14]	88.72
(8) DBDIN	89.03

6.5 Experiments on paraphrase identification

Quora We conducted experiments on Quora dataset to test the effectiveness of our model for paraphrase identification. We compared our model with the following published models on Quora dataset, and shown the results in Table 6.

The models (1) - (5) in Table 6 are sentence-encoding based methods without interaction. The Siamese-CNN model and Siamese-LSTM model encode sentences with CNN and LSTM respectively, and then predict the semantic relation between them based on the cosine similarity [1]. Multi-Perspective-CNN and Multi-Perspective-LSTM adopt multiple perspective cosine matching function [1]. Wang et al. [64] explore sentence similarity learning by lexical decomposition and composition (L.D.C). The models BiMPM [1] and AF-DMN [14] adopt interaction method with attention mechanism, and have shown improved performance over sentence-encoding based models. Specially, AF-DMN [14] shows that multi-layer neural network with attention mechanism can better extract interactive features for paraphrase identification.

As we can see, our single DBDIN outperforms the previous models and achieves 89.03% accuracy on the Quora test set. Therefore, the results further prove that our proposed model is also very effective to capture interactive features for paraphrase identification task.

7 Deep analysis and discussion

In this section, we gave in-depth analysis of model architecture and performed interpretable research for deep matching model. We first conducted an ablation study to investigate the effectiveness of the proposed components for model performance improvement. Then, we visualized the learned attentions and semantic representations for better understanding model behavior. Finally, we conducted case study and linguistic error analysis to investigate the matching results from the perspective of linguistics.

7.1 Ablation performance

We conducted an ablation study on DBDIN to examine the effectiveness of proposed cross sentence attention method, deep fusion and self-attention mechanism.

7.1.1 Effect of cross sentence attention

We first verified the effectiveness of the cross sentence attention as an essential component and shown the results in Table 7 (1). As mentioned before, we utilized the original sentence representation as the inputs of attention in

Table 7 Ablation study on SciTail dataset

Models	Dev	Test
(1) Effect of different attention strategies		
DBDIN (original-attention)	88.9	86.8
DBDIN (parallel-attention)	87.1	84.2
(2) Effect of the different number of cross sentence interaction unit		
1	86.9	84.0
2	88.6	85.6
3	88.9	86.8
4	89.1	87.2
5	89.2	87.4
(3) Effect of deep fusion and self-attention		
DBDIN	88.9	86.8
w/o Deep fusion	85.8	84.7
w/o Self-attention	88.1	85.8

each interaction unit. These operations make DBDIN has a comprehensive understanding of fine-grained semantic relations, and learns interactive representation at each interaction unit by searching the most relevant part with the consideration of another sentence. We compared two cross sentence attention strategies: the proposed attention that repeatedly attends to the original sentence representation of another one, and the parallel attention [14] that pays attention to the interactive representation of another sentence.

In this experiment, we replaced the proposed attention in DBDIN with parallel attention, in which each attention focuses on the interactive representation of another sentence and two interaction directions share the same attentive weights. As shown in Table 7 (1), the performance of DBDIN significantly decreases when replacing with parallel attention, which means the attention target is critical for extracting interactive features at each attention layer. It proves that the proposed attention with original sentence representation can improve matching performance by reducing attention error propagation in multi-layer network.

We further verified the effect of the different number of cross sentence interaction unit on performance, as shown in Table 7 (2). As we can see, with the number of interaction unit increases from 1 to 5, the performance increases on both the development set and the test set of SciTail dataset. We can conclude that the multiple interaction units are effective for improving matching performance. However, the increasing rate of accuracy will slow down with the increment of the number of interaction units. Moreover, the parameters will grow rapidly with the increment of interaction unit, and a large of number of parameters will increase model complexity for optimization. Because of computational cost, we just set the number of cross sentence interaction unit to 3 in our experiment.

7.1.2 Effect of deep fusion and self-attention mechanism

We tested the effectiveness of deep fusion and self-attention mechanism, as shown in Table 7 (3). For the model without deep fusion, we removed the deep fusion layer at each interaction unit, and the accuracy dropped by 2.1% on the test set of SciTail dataset. The results demonstrate that the deep fusion can effectively improve accuracy. It indicates that deep fusion has more powerful capability to aggregate and propagate semantic features for deep interaction.

For model without self-attention, we removed the final self-attention layer, and the accuracy was degraded to 85.8%. This indicates that global matching information captured by self-attention layer is also effective in improving performance. We come to a conclusion similar to the previous study [14] that global information is important, but our model has lower computing complexity by using one layer of self-attention rather than multiple layers.

7.2 Visualization analysis

Neural models have achieved state-of-the-art performance on sentence matching. Yet unlike traditional feature-based models that assign and optimize weights to varieties of human interpretable features (parts-of-speech, syntactic parse features etc.), the behavior of deep learning models is much less easily interpreted. Here, we explore multiple strategies to interpret how neural models can learn effective semantic features for sentence matching, which provides a reference for future model design. We employed visualization techniques [65–67] like attention and representation plotting to interpret model behavior for performance improvement.

7.2.1 Word alignment learned by attention

Previous work [1, 14–16] has shown that attention mechanism can greatly improve sentence matching performance by improving word alignment accuracy between two sentences. Our attention with original sentence representation allows one sentence to repeatedly focus on the most relevant information of another sentence at each attention. Thus, we could cautiously interpret the interactive results using our attentive weights. The attentive weights contain information about how two sentences are aligned. Here, we investigated the word alignment learned by attention, and visualized the attention results. We compared the proposed attention strategy that attends to the original sentence representation of another one, and the parallel attention that attends to the interactive representation of another one [14].

Given an instance from the test set of the SciTail dataset: {P: *all living cells have a plasma membrane that encloses their contents*. Q: *all types of cells are enclosed by a membrane*. The label y: Entailment.}. We investigated the results produced by DBDIN with 3 cross sentence interaction units $P(t) \rightarrow Q$ ($t \in \{1, 2, 3\}$) and 1 self-attention layer. We visualized the learned attention matrices for each attention layer.

Attention with Original Sentence Representation From the cross sentence attention results in Fig. 4, we observe that different attention layers have the ability to focus on the different parts of another sentence Q. In the first attention layer, the same or similar words in each sentence have a high correspondence. But the first attention layer may have erroneous alignments. We can find that the premise word "*encloses*" is incorrectly aligned to the hypothesis

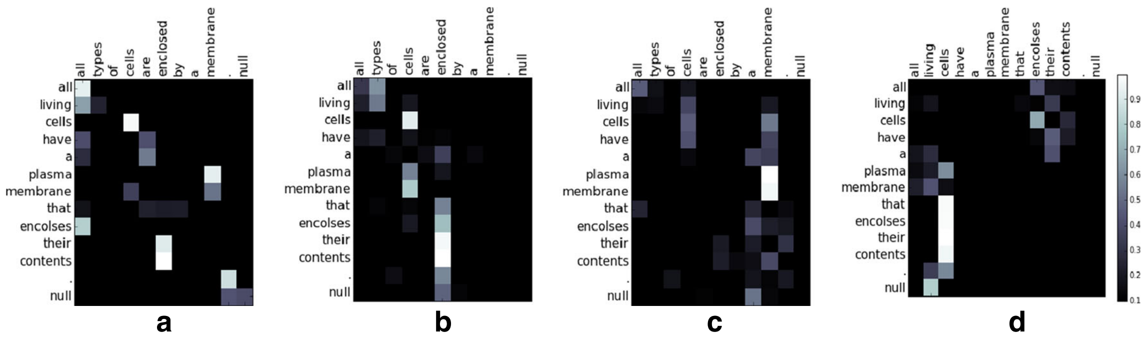


Fig. 4 The visualization of alignment matrices of the three cross sentence attention layers and the one self-attention layer. These results are produced by our proposed attention that attends to the original sentence representation of another one. **a** 1st cross attention. **b** 2nd cross attention. **c** 3rd cross attention. **d** self-attention

word "*all*". In the second attention layer, the alignment quality is improved dramatically, where the "*encloses*" is correctly aligned to "*enclosed*". It shows that the second attention layer effectively revises the errors from the first attention layer. Meanwhile, in the second and third attention layers, the attention gradually tends to capture phrase-level alignments, such as "*that encloses their contents*" and "*enclosed*", and "*cells have a plasma membrane*" and "*membrane*". With the increment of interaction units, the high attention layers also tend to obtain new alignment that is not captured in low attention layers. Judging by the aligned terms, the model is undoubtedly able to classify the label as an entailment, correctly.

In the self-attention layer, we observe that the phrase "*plasma membrane that encloses their contents*" is strongly aligned to the phrase "*living cells*". This indicates that the self-attention layer can capture global sentence-level relevance to enhance matching information within the sentence.

Attention with Interactive Sentence Representation To compare our proposed multi-layer attention with traditional attention method, we further analysed the results of parallel attention that is performed between two intermediate interactive layers [14]. The results in Fig. 5 are produced by the DBDIN with parallel attention. We observe that the first cross attention can capture some part of word alignments between the two sentences, but the second, third cross attentions and self-attention become unstable and ineffective for capturing word alignments. As a result, the higher attention layers can't capture more alignment information that is important for judging the semantic relation between the two sentences.

Additive Attention To verify the overall alignment quality of all attentions between the two sentences, we further performed an additive operation on the three cross sentence attention matrices, as shown in Fig. 6. As we can see, our proposed method attending to the original sentence representation of another one shows a more clear and accurate alignment, while the parallel attention with interactive representation is not capable of capturing some key alignment information between the two sentences.

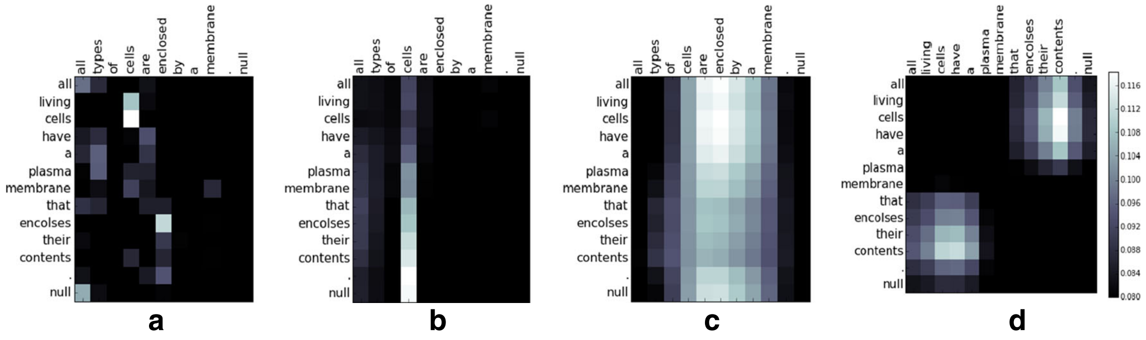


Fig. 5 The visualization of alignment matrices of the three cross sentence attention layers and the one self-attention layer. These results are produced by using parallel attention that attends to the interactive representation of another one. **a** 1st cross attention. **b** 2nd cross attention. **c** 3rd cross attention. **d** self-attention

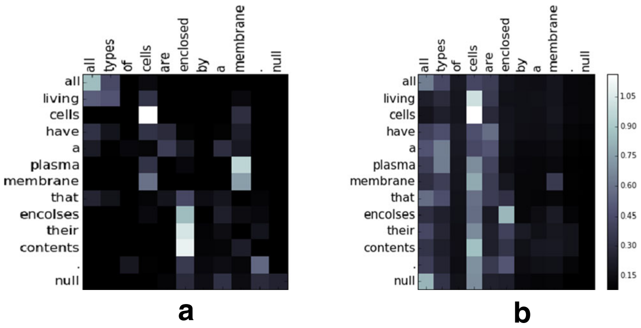


Fig. 6 The visualization of additive alignment matrices in the three cross sentence attention layers. **a** is the additive attention results from our proposed attention with original sentence representation. **b** is the results from parallel attention that attends to the interactive representation of another one

Finally, the visualized results of two types of attention verify that our proposed deep interaction network, equipped with attention with original sentence representation, deep fusion and self-attention, can accurately learn fine-grained semantic alignment between two sentence for improving sentence matching performance.

7.2.2 Semantic representation learned by deep interaction network

Furthermore, we explored the learned semantic representations in deep interaction network to analyze model behavior. Given representations H for input words with the associated gold class label c, the goal is to decide which units of H make the most significant contribution to the choice of class label c. Inspired by previous visualization techniques [65–67], we conducted visualization of layer-wise representation and layer-wise first-derivative saliency on each neural unit. The layer-wise representation is inspired by the forward-propagation strategy, which measures the learned semantic property values of each layer. The layer-wise first-derivative saliency is inspired by the back-propagation strategy, which measures how much each layer contributes to the final decision. Both of them assume that the larger value of input neural unit, the greater impact on the output.

Layer-Wise Representation Given the instance {P: *all living cells have a plasma membrane that encloses their contents*. Q: *all types of cells are enclosed by a membrane*. The label y: Entailment.}, we analysed the results of P, as shown in Fig. 7, where the 0 layer is word embedding, the 1 layer is original sentence representation with contextual information, the 2-4 layers are learned by cross sentence interaction unit, and the 5 layer is learned by self-attention. The darker point indicates the higher importance for the final decision. In our experiment, in order to facilitate implementation, we selected 50 dimension of each word representation and part of words for visualization.

We first visualized layer-wise representation of words "*cell*" and "*by*" , as shown in Fig. 7a and b. As we can see, with the increment of network depth, the semantic property values of word "*cell*" are larger than word "*by*", which indicates "*cell*" contributes more than "*by*" for learning the final semantic representation. It means that "*cell*" is a more important word than "*by*" in deciding the final semantic relation. The results show that different types of word have different importance to the final decision, and the functional word "*by*" is less important in this example.

To further verify the effect of deep fusion for semantic representation, we analyzed the model without deep fusion layer, as shown in Fig. 7c. When we removed the deep fusion, we can see that the semantic property values

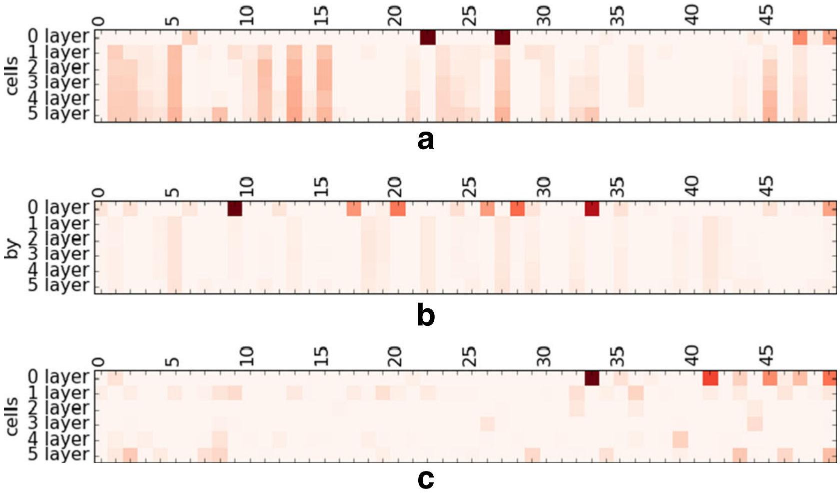


Fig. 7 The visualization of semantic representation in different layers. The darker point indicates that the corresponding value is greater. **a** The semantic representation of word cells in different layers. **b** The semantic representation of word by in different layers. **c** The semantic representation of word cells in different layers without deep fusion

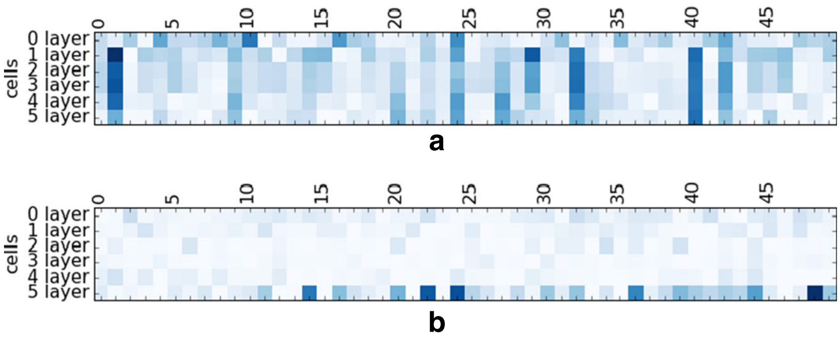


Fig. 8 The visualization of gradients in different layers. The darker point indicates that the corresponding value is greater. **a** The gradient of word cells in different layers. **b** The gradient of word cells in different layers without deep fusion

of "*cell*" tend to become smaller in higher layers. The results demonstrate that using deep fusion over deep matching network has more powerful capability retaining collected interactive features to learn sentence semantic representation, which plays a crucial role for improving matching performance in our deep interaction model.

Layer-Wise First-Derivative Saliency We then conducted another strategy to measure how much each input unit contributes to the final decision, which can be approximated by first derivatives [67]. We gave layer-wise first-derivative saliency for word "*cell*" as it is an important word for the final decision, as shown in Fig. 8a and b. Figure 8a shows that each layer has larger gradient value, which indicates these layers have a positive contribution to the final decision. Especially, we observe that the low layers still have a larger gradient, and therefore their semantic features are also influential for the final decision. Figure 8b is the results without deep fusion, we can see that the gradients of low layers tend to vanish and lost impact

Table 8 Example wins and losses on SciTail test dataset

ID	Premise	Hypothesis	BLEU	Gold	DBDIN	AF-DMN
A	it is an ore of lanthanum metal, along with monazite.	metals start out as ore.	0.0	E	E	N
B	over 100 strains of the bacterium that cause lyme disease have been identified in the united states.	lyme disease is caused by bacteria.	0.12	E	E	N
C	in between we had low pressure systems passing, creating stormy weather and waves up around 8 to 10 meters high.	you can expect stormy if a center of low pressure is moving your way.	0.15	E	E	N
D	the organism lives inside small intestinal cells, mainly in the small intestine.	in the body, chemical digestion mainly takes place in the small intestine.	0.50	N	N	E
E	morc affects the earliest stages of sperm production, or meiosis.	meiosis is part of the process of gametogenesis, which is the production of sperm and eggs.	0.12	N	N	E
F	acid-base indicators are substances that change color as a function of ph, usually over a range of 1 to 2 ph units.	if a substance has a ph value greater than 7, that indicates that it is base.	0.19	N	N	E
G	multiple tissue types compose organs and body structures.	a(n) organ is a structure that is composed of one or more types of tissues.	0.05	E	N	N
H	nonvascular plant; mosses, liverworts, and hornworts.	moss is best classified as a nonvascular plant.	0.12	E	N	N
I	the liver is divided into the right lobe and left lobes.	the gallbladder is near the right lobe of the liver.	0.45	N	E	E

We compared the proposed DBDIN model with AF-DMN. The E indicates entailment relation and the N indicates neutral relation between premise and hypothesis on the final decision. The result also verify that the deep fusion among the layers is important to help gradient flow, which relieves the vanishing gradient problem for training deep interaction network and therefore improves sentence matching performance.

7.3 Case study and linguistic error analysis

We investigated some examples from SciTail test dataset to demonstrate the ability of DBDIN for matching sentence pairs. Table 8 shows some wins and losses. We compared our proposed DBDIN with the representative AF-DMN [14]. To evaluate the influence of linguistic features between two sentences for semantic classification, we computed BLEU score [68] between each premise and hypothesis pair. The BLEU score measures how many words are shared between two sentences. It assumes the more overlapped words between them, the closer their semantics are. In our experiment, we used 1-gram BLEU score.

Examples A-C are entailment cases, where DBDIN has the ability to correctly recognize entailment relation while AF-DMN is insufficient. Each of these examples has low BLEU score. Thus, it is more difficult for models to recognize the entailment relation between them because of the absence of related semantic clues. The second set of examples D-F are neutral cases, where DBDIN is correct while AF-DMN is incorrect. Example D has high BLEU score, for which the models generally tend to identify the relation to entailment. Although examples E and F have low BLEU score, AF-DMN can't classify them neutral relation correctly. Finally, our proposed model has a better performance over these cases. It verifies that the proposed components, including attention with original sentence representation, deep fusion and self-attention, have a stronger ability to extract relevance semantics between two sentences, to improve sentence matching performance.

Examples G-I are cases that all models get wrong. Examples G and H are entailment relation, but they have low BLEU score. Meanwhile, the word orders and syntactic structures ("*compose*" and "*is composed of*") between the two sentences of G are also quite different. It causes models to failure recognizing the entailment relation between them. From these results, we can find that neural models may suffer from semantic gap problem and also be insufficient for capturing compositional structure that is often presented in sentence matching. Example I is neutral relation where the two sentences have high lexical overlap and also the similar word orders, which confuses models to misclassify a entailment class. On this case, despite the example is being marked as non-entail by human evaluators, the models classify them overwhelmingly as entailment. For examples G-I, we can see that there is a negative correlation between semantic relevance and lexical overlap. This indicates that the models are over-reliant word-level information and has limited ability to process compositional semantic information for these examples involving complex reasoning.

By the error analysis, we can find that it is still difficult for model to process some cases that involve complex semantic understanding. For these difficult cases, sentence semantics suffer from more the issues such as polysemy, ambiguity, as well as fuzziness, by which the model may need more inference information to distinguish the semantic relatedness to make the correct decision. To achieve further performance improvement, one possible solution is to introduce more linguistic information, such as introducing

Table 9 Model performance with different BLEU scores on SciTail test dataset

BLEU	R	Num	DBDIN	AF-DMN
[0, 0.1)	E	70	72.86	70.00
	N	334	93.71	90.72
[0.1, 0.2)	E	178	78.65	74.16
	N	457	90.59	87.31
[0.2, 0.3)	E	216	89.35	86.11
	N	315	87.94	81.90
[0.3, 0.4)	E	190	87.37	85.26
	N	129	76.74	73.64
[0.4, 0.5)	E	112	91.96	89.29
	N	40	82.50	80.00
[0.5, 1])	E	76	97.37	97.37
	N	9	77.78	77.78

R indicates the annotated semantic relation and Num indicates the number of sentence pairs in the corresponding group. E indicates entailment relation and N indicates neutral relation between premise and hypothesis syntactic information for semantic representation and incorporating external paraphrase database [69] to help better understanding the lexical and phrasal semantics. It is also helpful to construct adversarial training examples for model learning to process this case in which semantic relevance and lexical overlap have negative correlation. We consider them to the future work.

7.4 Statistical investigation based on lexical overlap

As shown in Section 7.3, linguistic features are important for sentence semantic matching. In order to better analyze the relevance of matching performance and lexical overlap, we gave a statistical investigation, and the results are shown in Table 9. We split the test set into different groups based on BLEU score, and computed the matching performance on each group. We compared the proposed DBDIN model with AF-DMN model [14]. From Table 9, we can see that our model shows better performance for both entail and non-entail classifications in each group test set. These results further show that our model has better ability to extract related semantic features for improving sentence matching performance.

Furthermore, we can see that the models tend to obtain a high accuracy for entailment relation on the cases with high BLEU score, and in reverse the models tend to obtain a high accuracy for neutral relation on the cases with low BLEU score. It is consistent with human judgment that the more lexical overlap between two sentences, the more possibility to be entailment relation. On the other hand, models present a low accuracy for the sentence pairs with low BLEU score but entailment relation, and high BLEU score but non-entailment relation. It indicates that there is still a lot of room for performance improvement on these extreme examples. For these examples, in the future, it will be helpful to introduce knowledge base to enhance lexical semantic matching, and also to explore better encoder architecture that is more sensitive to word orders.

8 Conclusions and future work

Within the attention-based interaction framework, we proposed an *Deep Bi-Directional Interaction Network* (DBDIN) which aims to better model the related semantic information between two sentences for sentence matching. We combined the advantages of attention and deep neural network to learn interactive features, apart from this, three novel features extraction methods: cross sentence attention with original sentence representation, deep fusion and self-attention mechanism, have been jointly presented in this paper. These methods benefit sentence matching model in the following three aspects:

1. The attention with original sentence representation allows the model is able to pay close attention to the relevant parts of another sentence, and therefore to learn more clear and accurate word alignments. The multiple interaction units allow one sentence to repeatedly read the information of another one, and therefore to better capture the related semantic information.
2. The combination of attention and deep fusion effectively retains semantic features learned at different interaction layers. As a result, it consequently improves semantic matching performance in deep interaction network.
3. The self-attention mechanism after the cross sentence interaction enhances global matching information, and further improves model performance.

We conducted experiments on two sentence matching tasks: natural language inference and paraphrase identification. Experimental results show that the proposed methods outperform the other methods with the three widely used evaluation datasets: SNLI, SciTail and Quora. By taking consideration of the above points, compared with traditional multiple-layer attention models, our methods can model sentence matching more precisely.

Furthermore, we conducted interpretable study to disclose how our deep interaction network with attention can benefit sentence matching, which provides a reference for future model design. We performed deep analyses with the proposed methods. The visualization results verify that our model is indeed able to capture more accurate word alignments than previous models, and the deep fusion can help model to learn effective semantic features in deep interaction network. The proposed method which inherit these advantages improves performance. Case study and linguistic error analysis reveal that the current models still have shortcomings in processing some extreme cases, and these analyses point out the direction for further performance improvement.

In the future, we will explore the encoder architecture that can better consider word orders to learn sentence representation. To improve this performance even further, it will be beneficial to study linguistic factors from various perspectives, e.g., syntactic structure, paraphrase database [69] and adversarial training examples, to help learning more accurate and robust sentence representation. Moreover, it also is meaningful to study a lightweight neural network model to combine pre-training techniques (such as pre-trained BERT [27]) with our model in the case of limited computing resources.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Mingtong Liu, Yujie Zhang, Jinan Xu and Yufeng Chen. The first draft of the manuscript was written by Mingtong Liu and

all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The research work descried in this paper has been supported by the National Nature Science Foundation of China (No. 61876198, 61976015, 61370130 and 61976016), and also supported by the Beijing Municipal Natural Science Foundation (No. 4172047), and the International Science and Technology Cooperation Program of China (No. K11F100010).

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Wang Z, Hamza W, Florian R (2017) Bilateral multi-perspective matching for natural language sentences. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. AAAI Press, pp 4144–4150
2. Bowman S, Angeli G, Potts C, Manning CD (2015) A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp 632–642
3. Iftene A, Balahur-Dobrescu A (2007) Hypothesis transformation and semantic variability rules used in recognizing textual entailment. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Association for Computational Linguistics, pp 125–130
4. Madnani N, Tetreault J, Chodorow M (2012) Re-examining machine translation metrics for paraphrase identification. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 182–190
5. Yin W, Schütze H, Xiang B, Zhou B (2016) Abcnn: Attention-based convolutional neural network for modeling sentence pairs. Trans Assoc Comput Linguist 4:259–272
6. Clark P, Etzioni O, Khot T, Sabharwal A, Tafjord O, Turney P, Khashabi D (2016) Combining retrieval, statistics, and inference to answer elementary science questions. In: Thirtieth AAAI Conference on Artificial Intelligence
7. Esposito M, Damiano E, Minutolo A, De Pietro G, Fujita H (2020) Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering. Inf Sci 514:88–105
8. Xiao L, Wissmann D, Brown M, Jablonski S (2004) Information extraction from the web: System and techniques. Appl Intell 21(2):195–224
9. Androutsopoulos I, Malakasiotis P (2010) A survey of paraphrasing and textual entailment methods. J Artif Intell Res 38:135–187
10. Liu Q, Huang Z, Huang Z, Liu C, Chen E, Su Y, Hu G (2018) Finding similar exercises in online education systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 1821–1830
11. OShea K (2012) An approach to conversational agent design using semantic sentence similarity. Appl Intell 37(4):558–568
12. Gong Y, Luo H, Zhang J (2017) Natural language inference over interaction space. arXiv:1709.04348
13. Liu P, Qiu X, Chen J, Huang X (August 2016) Deep fusion LSTMs for text semantic matching. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin
14. Duan C, Cui L, Chen X, Wei F, Zhu C, Zhao T (2018) Attention-fused deep matching network for natural language inference. In: IJCAI, pp 4033–4040
15. Parikh A, Täckström O, Das D, Uszkoreit J (2016) A decomposable attention model for natural language inference. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp 2249–2255
16. Rocktäschel T, Grefenstette E, Hermann KM, Kočiskỳ T, Blunsom P (2015) Reasoning about entailment with neural attention. arXiv:1509.06664
17. Park C, Song H, Lee C (2020) S3-net: Sru-based sentence and self-matching networks for machine reading comprehension. ACM Trans Asian Low-Resource Lang Inf Process (TALLIP) 19(3):1–14
18. Peng D, Wu S, Liu C (2019) Mpsc: A multiple-perspective semantics-crossover model for matching sentences. IEEE Access 7:61320–61330
19. Pota M, Esposito M, Pietro GD, Fujita H (2020) Best practices of convolutional neural networks for question classification. Appl Sci 10(14):4710
20. Tan M, Santos CD, Xiang B, Zhou B (2015) Lstm-based deep learning models for non-factoid answer selection. arXiv:1511.04108
21. Chen Q, Zhu X, Ling Z-H, Wei S, Jiang H, Inkpen D (2017) Enhanced lstm for natural language inference. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 1657–1668
22. Ghaeini R, Hasan SA, Datla V, Liu J, Lee K, Qadir A, Ling Y, Prakash A, Fern X, Farri O (2018) Dr-bilstm: Dependent reading bidirectional lstm for natural language inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp 1460–1469
23. Bjerva J, Plank B, Bos J (2016) Semantic tagging with deep residual networks. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp 3531–3541
24. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
25. Heilman M, Smith NA (2010) Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, pp 1011–1019
26. Wang Z, Ittycheriah A (2015) Faq-based question answering via word alignment. arXiv:1507.02628
27. Devlin J, Chang M-W, Lee K, Toutanova K (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp 4171–4186
28. Mou L, Men R, Li G, Xu Y, Zhang L, Yan R, Jin Z (2016) Natural language inference by tree-based convolution and heuristic matching. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp 130–136
29. Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp 2227–2237

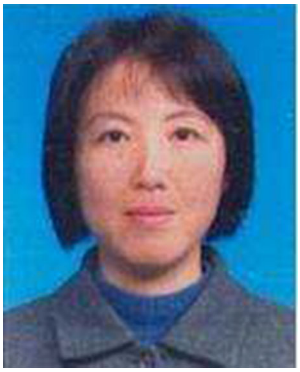
30. Wang S, Jiang J (2016) A compare-aggregate model for matching text sequences. arXiv:1611.01747
31. Tai KS, Socher R, Manning CD (2015) Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp 1556–1566
32. Bowman S, Gauthier J, Rastogi A, Gupta R, Manning CD, Potts C (2016) A fast unified model for parsing and sentence understanding. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 1466–1477
33. Hermann KM, Kocisky T, Grefenstette E, Espeholt L, Kay W, Suleyman M, Blunsom P (2015) Teaching machines to read and comprehend. In: Advances in Neural Information Processing Systems, pp 1693–1701
34. Wang S, Jiang J (2016) Learning natural language inference with lstm. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 1442–1451
35. Yang R, Zhang J, Gao X, Ji F, Chen H (July 2019) Simple and effective text matching with richer alignment features. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence
36. Im J, Cho S (2017) Distance-based self-attention network for natural language inference. arXiv:1712.02047
37. Lin Z, Feng M, Santos CND, Yu M, Xiang B, Zhou B, Bengio Y (2017) A structured self-attentive sentence embedding. arXiv:1703.03130
38. Chen Q, Zhu X, Ling Z-H, Inkpen D, Wei S (2018) Neural natural language inference models enhanced with external knowledge. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 2406–2417
39. Pan B, Yang Y, Zhao Z, Zhuang Y, Cai D, He X (2018) Discourse marker augmented network with reinforcement learning for natural language inference. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 989–999
40. Wang Y, Wang M, Fujita H (2020) Word sense disambiguation: A comprehensive knowledge exploitation framework. Knowl-Based Syst 190:105030
41. Miller, George A (1995) Wordnet: a lexical database for english. Commun ACM 38(11):39–41
42. Pennington J, Socher R, Manning C (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
43. Chomsky N (1957) Syntactic structures. the hague: Mouton.. 1965. aspects of the theory of syntax. Cambridge, Mass.: MIT Press.(1981) Lectures on Government and Binding, Dordrecht: Foris. (1982) Some Concepts and Consequences of the Theory of Government and Binding. LI Monographs, vol 6, p 1–52
44. Dowty D (2007) Compositionality as an empirical problem. Direct Compositional (14):23–101
45. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
46. Dozat T, Manning CD (2016) Deep biaffine attention for neural dependency parsing. arXiv:1611.01734
47. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp 315–323
48. Fan H, Zhou J (2018) Stacked latent attention for multimodal reasoning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
49. Srivastava RK, Greff K, Schmidhuber J (2015) Highway networks. arXiv:1505.00387
50. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166
51. Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp 1724–1734
52. Tay Y, Tuan LA, Hui SC (2017) A compare-propagate architecture with alignment factorization for natural language inference. arXiv:1801.00102
53. Kingma D, Ba J (2014) Adam: A method for stochastic optimization. arXiv:1412.6980
54. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp 448–456
55. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
56. Khot T, Sabharwal A, Clark P (2018) Scitail: A textual entailment dataset from science question answering. In: Thirty-Second AAAI Conference on Artificial Intelligence
57. Kim S, Kang I, Kwak N (2019) Semantic sentence matching with densely-connected recurrent and co-attentive information. Proc AAAI Conf Artif Intell 33:6586–6593. https://doi.org/10.1609/aaai.v33i01.33016586
58. Cheng J, Dong L, Lapata M (2016) Long short-term memory-networks for machine reading. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp 551–561
59. Sha L, Chang B, Sui Z, Li S (2016) Reading and thinking: Re-read lstm unit for textual entailment recognition. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp 2870–2879
60. Paria B, Annervaz KM, Dukkipati A, Chatterjee A, Podder S (2016) A neural architecture mimicking humans end-to-end for natural language inference. arXiv:1611.04741
61. Zhang Z, Wu Y, Li Z, Zhao H (2019) Explicit contextual semantics for text comprehension. In: Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33)
62. Zhang Z, Wu Y, Zhao H, Li Z, Zhang S, Zhou X, Zhou X (2019) Semantics-aware bert for language understanding. arXiv:1909.02209
63. Yin W, Roth D, Schütze H (2018) End-task oriented textual entailment via deep explorations of inter-sentence interactions. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp 540–545
64. Wang Z, Mi H, Ittycheriah A (2016) Sentence similarity learning by lexical decomposition and composition. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp 1340–1349
65. Ding Y, Liu Y, Luan H, Sun M (2017) Visualizing and understanding neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 1150–1159
66. Karpathy A, Johnson J, Fei-Fei L (2015) Visualizing and understanding recurrent networks. arXiv:1506.02078
67. Li J, Chen X, Hovy E, Jurafsky D (2016) Visualizing and understanding neural models in nlp. In: Proceedings of the 2016 Conference of the North American Chapter of the Association

for Computational Linguistics: Human Language Technologies, pp 681–691

68. Papineni K, Roukos S, Ward T, Zhu WJ (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of Meeting of the Association for Computational Linguistics
69. Ganitkevitch J, Van Durme B, Callison-Burch C (2013) Ppdb: The paraphrase database. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 758–764

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mingtong Liu received a B.E. degree in computer science and information technology from Beijing Jiaotong University, Beijing, China, in 2016. He is pursuing a Ph.D. at Beijing Jiaotong Univerisity. His research focuses on dependency parsing, sentence matching, paraphrase generation, machine translation, and natural language processing.



Yujie Zhang received a B.E. degree in computer science and information technology from Beijing Jiaotong University, Beijing, China, in 1983 and a Ph.D. in computer science from the University of Electro-Communications, Tokyo, Japan, in 1999. She joined Beijing Jiaotong University in April 2011 and is currently a professor at the School of Computer and Information Technology. Her current research interests include machine translation, multilingual information processing, and natural language processing.

Jinan Xu received a B.E. degree from the Communication and Control Engineering Department from Northern Jiaotong University, Beijing, China, in 1992 and M.S. and Ph.D. degrees from Hokkaido University, Japan, in 2003 and 2006 respectively. From 2006 to 2009, he worked as an expert researcher at the Central Research Institute of Nippon Electronic Company (NEC). He joined Beijing Jiaotong University in 2009 and is currently a professor at the School of Computer and Information Technology. His research interests include natural language processing, machine translation, AI, and affective computing.

Yufeng Chen received an undergraduate degree in mechanical electrical engineering from Beijing Jiaotong University in 2003, and her Ph.D. in pattern recognition and intelligent systems from the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, in June 2008. From July 2008 to September 2014, she joined NLPR and worked as an associate professor. Since September 2014, she has been working as an associate professor at the School of Computer and Information Technology, Beijing Jiaotong University. Her research interests include natural language processing, machine translation, and information retrieval.