RI-Match: Integrating Both Representations and Interactions for Deep Semantic Matching

Lijuan Chen[1,2](✉), Yanyan Lan[1,2], Liang Pang[1,2], Jiafeng Guo[1,2], Jun Xu[1,2], and Xueqi Cheng[1,2]

[1] CAS Key Lab of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
chenlijuan@software.ict.ac.cn,
{lanyanyan,pangliang,guojiafeng,junxu,cxq}@ict.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China

Abstract. Existing deep matching methods can be mainly categorized into two kinds, i.e. representation focused methods and interaction focused methods. Representation focused methods usually focus on learning the representation of each sentence, while interaction focused methods typically aim to obtain the representations of different interaction signals. However, both sentence level representations and interaction signals are important for the complex semantic matching tasks. Therefore, in this paper, we propose a new deep learning architecture to combine the merits of both deep matching approaches. Firstly, two kinds of word level matching matrices are constructed based on word identities and word embeddings, to capture both exact and semantic matching signals. Secondly, a sentence level matching matrix is constructed, with each element stands for the interaction between two sentence representations at corresponding positions, generated by a bidirectional long short term memory (Bi-LSTM). In this way, sentence level representations are well captured in the matching process. The above matrices are then fed into a spatial recurrent neural network (RNN), to generate the high level interaction representations. Finally, the matching score is produced by a k-Max pooling and a multilayer perceptron (MLP). Experiments on paraphrasing identification shows that our model outperforms traditional state-of-the art baselines significantly.

Keywords: Deep semantic matching · Word level interactions
Sentence level representations

1 Introduction

Matching two sentences is a core problem of many applications in natural language processing, such as information retrieval and question answering. Taking information retrieval as an example, given a query and a document, a matching

Y.-H. Tseng et al. (Eds.): AIRS 2018, LNCS 11292, pp. 90–102, 2018.
https://doi.org/10.1007/978-3-030-03520-4_9

function is created to determine the relevance degree between the query and the document.

Recently, deep neural networks have been applied in this area and achieved some progresses. These methods can be mainly categorized into two kinds: representation focused methods and interaction focused methods. Representation focused methods first encode each sentence as one dense vector, and then calculate the similarities of two sentences vectors as the matching score. Typical examples include ARC-I [3] and CNTN [8]. In general, this approach is straightforward and capable to capture the high level semantic meanings of each sentence. However, it will miss important detailed information by compressing such an entire sentence into a single vector. To tackle this problem, interaction focused methods turn to directly learn the interactions between two sentences. They first construct a word level matching matrix to capture detailed word level interaction signals. Then a deep neural network is applied on this matrix to abstract high level interaction signals. Finally, an MLP is used to calculate the matching score. State of the art methods include ARC-II [3], MatchPyramid [7], Match-SRNN [12], MV-LSTM [11], and BiMPM [14]. Interaction focused methods have the ability to integrate rich interaction signal, however, sentence level semantic meanings are not fully captured.

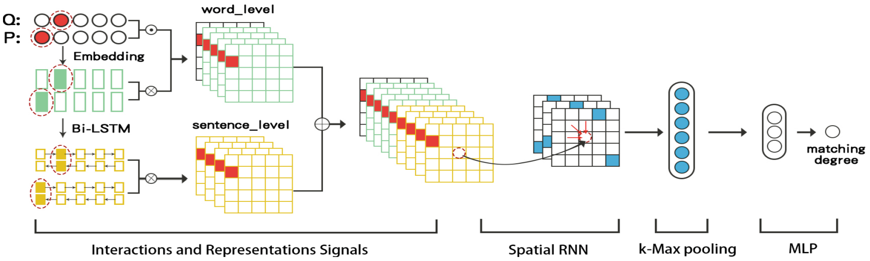

Fig. 1. An overview of RI-Match. ⊙(interaction in the word identities). ⊗(all interaction functions partly defined in word embedding level and sentence level).

Semantic matching is such a complex problem that both interaction signals and sentence representations need to be considered. In this paper, we propose a new deep architecture to integrate them, namely RI-Match[1]. For word level, we capture the interaction signals by the word identity and word embedding. For sentence level information, we adopt a Bi-LSTM to scan each sentence, and the sentence representations are obtained. Then a matrix can be constructed by computing the similarity between two sentence representations at corresponding positions. Finally, the above matrices are fed into a spatial RNN [12], which captures both nearly and long distant interactions. Furthermore, a k-Max pooling strategy [11] is adopted to select the top k strongest interaction signals, and a multi-layer perceptron(MLP) is utilized to obtain the matching score.

[1] We will release our code when our paper accepted.

We conduct two experiments on different tasks, such as paraphrasing identification and answer selection. The experimental results show that RI-Match outperforms traditional interaction and representation focused methods on paraphrasing identification tasks, demonstrating the advantage of combining the merits of both approaches.

2 Related Work

Existing deep learning methods for semantic matching can be mainly categorized into two kinds, i.e representation focused methods and interaction focused methods.

Representation focused methods represent two input sentences individually to a dense vectors in the same embedding space, and then define different functions to calculate the matching degree of the two sentence vectors. It is common to adopt the recursive neural network. The advantage of this method is to model complex and semantic phenomenon in the sentence level. It is easy to complete. However, the disadvantage is that the encoder would loss detail information of texts pair.

To tackle this problem, interaction focused methods has been proposed. It turns to capture more interactions relationship between two texts in word level. Then the matching degree can be determined by interaction matrix, which has achieved much attention, examples include MatchPyramid, MatchSRNN and MV-LSTM. Our method combines the merits of both deep matching methods.

The MatchPyramid constructs basic word level interaction matrix by defining different similarity functions. Then the model regards the semantic matching as image recognition by considering the matching matrix as one image. For this method, it will lose complex semantic matching information in the word level.

To overcome above defects, the same in the word level interaction, the MatchSRNN adopts neural tensor network to capture more complicated interactions [9]. Then the spatial RNN calculate the matching degree by the above interactions. This method partly solves the single matching matrix problem and output an interaction tensor. However, it cannot obtain the higher level interactions.

Different from the MatchPyramid and MatchSRNN, the MV-LSTM captures the interaction signals of texts in the sentence representations level. However, it is natural that the key words of texts sometimes partly determine the matching degree of sentences, this method cannot perfectly capture these signals.

For the above related work, it is natural for us to compose the semantic matching model with the different level signals, which takes both word level interaction signals and sentence level representations signals into account, namely RI-Match. It can richly capture the complex semantic matching information by feeding these two kinds level signals to next layer.

3 Method

In this section, we introduce our method which integrates both representation and interaction signals for deep semantic matching, namely RI-Match. As shown in Fig. 1, RI-Match consists of four components.

3.1 Interactions and Representations Structure

The goal of this component is to construct matching signals for two sentences in word interaction level and sentence representations level. Given two sentence $Q = (q_1, \cdots, q_m)$ and $P = (p_1, \cdots, p_n)$, where q_i and p_j denotes the i-th and j-th word in sentence Q and P. Sequence of word embeddings can be obtained by mapping each word identity into a vector, where we have $\mathbf{Q} = (\boldsymbol{q}_1, \cdots, \boldsymbol{q}_m)$ and $\mathbf{P} = (\boldsymbol{p}_1, \cdots, \boldsymbol{p}_n)$. In order to construct different level matching signals, we input both word identities Q and P, and word embeddings $\mathbf{Q}$ and $\mathbf{P}$.

Word Level Interactions Signal
The goal of this part is to represent word level interaction signals for sentences based on word identities and word embedding. We present several matching signals as a matching matrix $\mathbf{M}$, with each word-pair s_{ij} express the basic interaction between word q_i and p_j.

Based on Word Identities. It is natural for us to think that two sentences are more relevant if they contain more identical words.

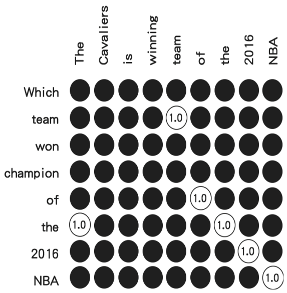

Fig. 2. Xnor operator, where the black circle elements are all value 0.

Xnor can capture such information as follows,

$$s_0(q_i, p_j) = q_i \odot p_j. \tag{1}$$

where the $\odot$ stands for the $Xnor$ operation, which produce either 1 or 0 to measure whether two words are the same. We can visualize the matching matrix $\mathbf{M_0}$ in Fig. 2. As the example shows that both Q and P contain the words {*team, 2016, NBA*}, these partly determine the matching degree of two sentences.

However, the disadvantage of this similarity operator is that it cannot obtain the semantic matching situations, for example, the similar words matching signals are ignore. As we know, words ‘won’ and ‘winning’ have the similar meaning than words ‘won’ and ‘2016’. In order to capture word semantic similarities, we define similarity operators based on the word embeddings.

Based on Word Embedding. The word embedding is a fixed vector for every individual word, which is pre-trained by Glove. $\boldsymbol{q}_i$ and $\boldsymbol{p}_j$ stand for the i-th and j-th word representations in sentence Q and P.

Cosine is a common way to calculate similarity of two word embeddings, which regarded as the angle of two vectors. We show it as follows,

$$s_1(\boldsymbol{q}_i, \boldsymbol{p}_j) = \frac{\boldsymbol{q}_i^T \boldsymbol{p}_j}{\|\boldsymbol{q}_i\| \cdot \|\boldsymbol{p}_j\|}. \tag{2}$$

where $\| \cdot \|$ stands for the norm of the vector, we adopt L2 norm in this paper. Cosine function guarantees that exact matching signals will get the highest similarity scores 1.

Dot Product compares to cosine similarity operator, it takes the norm of word embeddings into account.

$$s_2(\boldsymbol{q}_i, \boldsymbol{p}_j) = \boldsymbol{q}_i^T \boldsymbol{p}_j. \tag{3}$$

The norm of word embedding can be interpreted as the importance of the word, for example, none word ‘NBA’ should be important than empty word ‘the’.

Both cosine similarity and dot product similarity treat word similarity as a scale value s_1 and s_2 and obtain the matching matrix $\mathbf{M_1}$ and $\mathbf{M_2}$ shown in Fig. 3 while we also treat word similarity as a representation, such as a vector.

Element-Wise Multiplication is a direct way to combine the signals for two word vectors, and output a similarity representation of two words, which can be represented as follows,

$$\boldsymbol{s_3}(\boldsymbol{q}_i, \boldsymbol{p}_j) = \boldsymbol{q}_i \odot \boldsymbol{p}_j. \tag{4}$$

where $\odot$ stands for element-wise multiplication. This method can produce the interaction matching vector matrix $\mathbf{M_3}$ which is different from the cosine and dot product, since they just obtain interaction matrix. This interaction signal can largely retain more details of sentences based on the word embedding.

Word Meaning Signal. Many deep models describe the word level interaction signals mostly in the interaction of two sentences. However, the meaning of word itself contains much useful information for semantic matching. For this purpose, we concatenate this signal with the above-mentioned word level interaction signals together to enrich the word level signals. First, we use the RNN to compress the dimension for every word embedding of each sentence, which can transform the word embedding to a dense vector in low dimension. Meanwhile, it can avoid a great difference between diverse signals in the dimension. It can

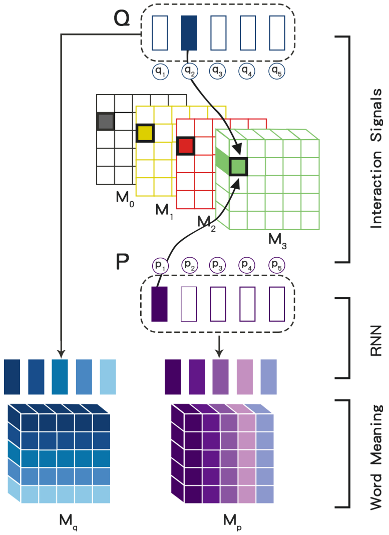

Fig. 3. Word level signals, where the matrix M_0, M_1, M_2, M_3 respectively are the result with operator of xnor, cosine, dot product and the element-wise multiplication.

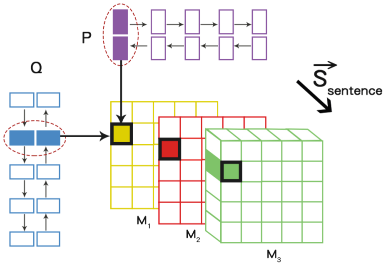

Fig. 4. Sentence level representations signals, where the matrix M_1, M_2, M_3 respectively are the result with operator of cosine, dot product and the element-wise multiplication.

also enrich the representations of every word. For the sentence $\mathbf{Q} = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m)$ and $\mathbf{P} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n)$, it will get the new word representations as follows.

$$(\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m) \overset{RNN}{\longrightarrow} (\boldsymbol{q}_1^l, \ldots, \boldsymbol{q}_m^l),$$
$$(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n) \overset{RNN}{\longrightarrow} (\boldsymbol{p}_1^l, \ldots, \boldsymbol{p}_n^l).$$

where the $\boldsymbol{q}_i^l$ and $\boldsymbol{p}_j^l$ separately stands for the new word representations for q_i and p_j, we show the RNN state in Fig. 3.

We cannot directly concatenate above representations with the word interaction signals. In the word interaction signals, it contains the $m \times n$ vectors, each vector represents every word-pairs interactions of two sentences. However for $\mathbf{Q}$ it contains m vectors and for $\mathbf{P}$ it contains n vectors. In order to concatenate word meaning signals with interaction signals, we stack every new word representations for n times to get the vector matrix M_q. For this method, We can get the vector matrix $\mathbf{M_p}$ in the similar way for m times as follows.

$$\boldsymbol{s}_q(ij), \forall j \in [1, n] \Leftrightarrow \boldsymbol{q}_i^l$$

$$\boldsymbol{s}_p(ij), \forall i \in [1, m] \Leftrightarrow \boldsymbol{p}_j^l$$

where the $\boldsymbol{s}_q(ij)$ is an vector in $\mathbf{M_q}$ shown in Fig. 3, $\boldsymbol{q}_i^l$ stacked with n times by row with the same color, therefore when $\forall j \in [1, n]$, they all stands for the $\boldsymbol{q}_i^l$. Similarly, we obtain $\mathbf{M_p}$ by stacking $\boldsymbol{p}_j^l$ with m times by column.

Finally, by concatenating the above-mentioned signals based on word identities and word embedding, we get the signals vector of sentences in word level as follows. We can see Fig. 3, it shows the detail of word level signals including word interactions signals and word meaning signals.

$$\boldsymbol{s}_{word} = [s_0, s_1, s_2, \boldsymbol{s}_3^{\mathrm{T}}, \boldsymbol{s}_q^{\mathrm{T}}, \boldsymbol{s}_p^{\mathrm{T}}]^{\mathrm{T}} \tag{5}$$

Sentence Level Representations Signal

The higher level matching signals are important to determine whether two sentences are match. The sentence representations signals depend on contextual information. Therefore, we adopt a parameter-shared bi-directional LSTM [4] to encode contextual embeddings to capture such information. It can well capture nearby words in the encode process.

For embedding matrix $\mathbf{Q} = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m)$, bidirectional LSTM takes both previous and future context into account from two directions. Therefore, we utilize Bi-LSTM to process the input for each time-step as follows:

$$\begin{aligned}\overrightarrow{h}_i^q &= \overrightarrow{LSTM}(\boldsymbol{h}_{i-1}^q, \boldsymbol{q_i}), i \in [1, m],\\ \overleftarrow{h}_i^q &= \overleftarrow{LSTM}(\boldsymbol{h}_{i+1}^q, \boldsymbol{q_i}), i \in [1, m].\end{aligned} \tag{6}$$

Meanwhile, we apply the same BiLSTM to encode $\mathbf{P} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n)$:

$$\begin{aligned}\overrightarrow{h}_j^p &= \overrightarrow{LSTM}(\boldsymbol{h}_{j-1}^p, \boldsymbol{p_j}), j \in [1, n],\\ \overrightarrow{h}_j^p &= \overleftarrow{LSTM}(\boldsymbol{h}_{j+1}^p, \boldsymbol{p_j}), j \in [1, n].\end{aligned} \tag{7}$$

Therefore, we concatenate two vectors $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ together as $\boldsymbol{h} = [\overrightarrow{h}_t^{\mathrm{T}}, \overleftarrow{h}_t^{\mathrm{T}}]^{\mathrm{T}}$ for each position of sentence. It stands for t-th sentence representations from the two directions of the whole sentence. The $(\cdot)^T$ stands for the transposition operation. Then we can obtain the encode matrices as $\mathbf{Q_{Encode}} = (\boldsymbol{h}_1^q, \ldots, \boldsymbol{h}_m^q)$ and $\mathbf{P_{Encode}} = (\boldsymbol{h}_1^p, \ldots, \boldsymbol{h}_n^p)$ We adopt the interaction functions used in word level interaction signals, then we can obtain the sentence representations interaction vector as follows, the Fig. 4 shows the detail of sentence representations signals vector.

$$\begin{aligned}\mathbf{s_1}(\boldsymbol{h}_i^q, \boldsymbol{h}_j^p) &= \frac{\boldsymbol{h}_i^{qT}\boldsymbol{h}_j^p}{\|\boldsymbol{h}_i^q\| \cdot \|\boldsymbol{h}_j^p\|},\\ \mathbf{s_2}(\boldsymbol{q}_i, \boldsymbol{p}_j) &= \boldsymbol{h}_i^{qT}\boldsymbol{h}_j^p.\\ \boldsymbol{s_3}(\boldsymbol{h}_i^q, \boldsymbol{h}_j^p) &= \boldsymbol{h}_i^q \odot \boldsymbol{h}_j^p,\\ \boldsymbol{s}_{sentence} &= [\mathbf{s_1}, \mathbf{s_2}, \boldsymbol{s_3}^{\mathrm{T}}]^{\mathrm{T}}.\end{aligned} \tag{8}$$

Finally, we concatenate the word level interactions signals and sentence level representations signals together as follow.

$$\boldsymbol{s}_{ij} = [\boldsymbol{s}_{word}^{\mathrm{T}}, \boldsymbol{s}_{sentence}^{\mathrm{T}}]^{\mathrm{T}}. \tag{9}$$

where $\boldsymbol{s}_{ij}$ contains multiple signals in word interactions level and sentence representations level. For word level, it contains the word meaning signals and word interaction signals. Therefore, the output of this layer is tensor matrix of signals.

3.2 Spatial RNN

The second step is to apply spatial RNN to obtain the sentence interaction from multi-signals construction layer, from which we get multiple signals of two sentences in the word level and sentence representations level. Spatial RNN is a variation on the multi-dimensional RNN [1]. For the spatial RNN, given the interactions representations of prefixes $Q[1:i-1] \sim P[1:j], Q[1:i] \sim P[1:j-1]$ and $Q[1:i-1] \sim P[1:j-1]$, expressed as $\boldsymbol{h}_{i-1,j}, \boldsymbol{h}_{i,j-1}$ and $\boldsymbol{h}_{i-1,j-1}$, the interaction of prefixes $Q[1:i] \sim P[1:j]$ can be calculated by following equation:

$$\boldsymbol{h}_{ij} = f(\boldsymbol{h}_{i-1,j}, \boldsymbol{h}_{i,j-1}, \boldsymbol{h}_{i-1,j-1}, \boldsymbol{s}_{ij}). \tag{10}$$

where $\boldsymbol{s}_{ij}$ stands for the signals information from the multi-signals construction layer including word level interactions signals and sentence level representations signals.

We have many choices for function f. We adopt GRU since it has shown excellent performance in many tasks. In this paper, we use spatial RNN changed from traditional GRU. We extend it to spatial RNN as follows.

$$\begin{aligned}
&\boldsymbol{q} = [\boldsymbol{h}_{i-1,j}^{\mathrm{T}}, \boldsymbol{h}_{i,j-1}^{\mathrm{T}}, \boldsymbol{h}_{i-1,j-1}^{\mathrm{T}}, \boldsymbol{s}_{ij}^{\mathrm{T}}]^{\mathrm{T}}, \\
&\boldsymbol{r}_l = \sigma(W^{(rl)}\boldsymbol{q} + \boldsymbol{b}^{(rl)}), \boldsymbol{r}_t = \sigma(W^{(rt)}\boldsymbol{q} + \boldsymbol{b}^{(rt)}), \\
&\boldsymbol{r}_d = \sigma(W^{(rd)}\boldsymbol{q} + \boldsymbol{b}^{(rd)}), \boldsymbol{r}^{\mathrm{T}} = [\boldsymbol{r}_l^{\mathrm{T}}, \boldsymbol{r}_t^{\mathrm{T}}, \boldsymbol{r}_d^{\mathrm{T}}], \\
&\boldsymbol{z}'_i = (W^{(zi)}\boldsymbol{q} + \boldsymbol{b}^{(zi)}), \boldsymbol{z}'_l = (W^{(zl)}\boldsymbol{q} + \boldsymbol{b}^{(zl)}), \\
&\boldsymbol{z}'_t = (W^{(zt)}\boldsymbol{q} + \boldsymbol{b}^{(zt)}), \boldsymbol{z}'_d = (W^{(zd)}\boldsymbol{q} + \boldsymbol{b}^{(zd)}), \\
&[\boldsymbol{z}_i, \boldsymbol{z}_l, \boldsymbol{z}_t, \boldsymbol{z}_z] = \mathrm{SoftmaxByRow}([\boldsymbol{z}'_i, \boldsymbol{z}'_l, \boldsymbol{z}'_t, \boldsymbol{z}'_z]),
\end{aligned} \tag{11}$$

$$\begin{aligned}
\boldsymbol{h}'_{ij} &= \phi(W\boldsymbol{s}_{ij} + U(\boldsymbol{r} \odot [\boldsymbol{h}_{i-1,j}^{\mathrm{T}}, \boldsymbol{h}_{i,j-1}^{\mathrm{T}}, \boldsymbol{h}_{i-1,j-1}^{T}]^{\mathrm{T}}) + \boldsymbol{b}), \\
\boldsymbol{h}_{ij} &= \boldsymbol{z}_l \odot \boldsymbol{h}_{i,j-1} + \boldsymbol{z}_{\boldsymbol{t}} \odot \boldsymbol{h}_{i-1,j} + \boldsymbol{z}_{\boldsymbol{d}} \odot \boldsymbol{h}_{i-1,j-1} + \boldsymbol{z}_{\boldsymbol{i}} \odot \boldsymbol{h}'_{i,j}.
\end{aligned} \tag{12}$$

where $U, W's$ and $b's$ are parameters, and SoftmaxByRow is a function to calculate softmax for every dimension by the four gates, as following:

$$[\boldsymbol{z}'_p]_j = \frac{e^{[\boldsymbol{z'}_p]_j}}{e^{[\boldsymbol{z'}_p]_j} + e^{[\boldsymbol{z'}_p]_l} + e^{[\boldsymbol{z'}_p]_t} + e^{[\boldsymbol{z'}_p]_d}}, p = i, l, t, d.$$

3.3 k-Max Pooling

Since spatial RNN getting the global interaction vectors between two texts, we introduce the third step to process such information by k-Max pooling.

The strong signals largely determine the matching degree of two sentences, these method has been approved to be valid in MV-LSTM. Therefore, we use k-Max pooling to automatically select top k strongest signals in the global interaction tensor, similar to [5]. Specifically for the spatial RNN matrix, we scan the matrix and directly return the top k values of every slice by the descending order to form a vector q.

3.4 MultiLayer Perception

Finally, we use a MLP to obtain the matching degree by aggregating the strong interaction information chosen by k-Max pooling, such information can be represented as vector q. For obtaining higher level representation r, vector q is feed into a full connection hidden layer. The final matching score can be obtained with a linear function:

$$r = f(W_s q + b_r), \quad s = W_s r + b_s.$$

where W_r and W_s donate the parameter metrics, and b_r and b_s are corresponding biases.

4 Experiments

In this section, we verify our model performance on two tasks: paraphrasing identification (PI) and answer sentence selection (ASS). We compare our model with state-of-the-art models on some standard benchmark datasets including Quora-question-pairs and WikiQA, to demonstrate the superiority of RI-Match against baselines.

4.1 Experimental Settings

First, we introduce our experimental settings, including parameter setting, and evaluation metrics.

Parameter Settings. We initialize word embeddings in the word embedding layer with 300-dimensional Glove word vectors pre-trained in the 840B Common Crawl corpus. On the paraphrasing identification task, we set the hidden dimension as 50 for Bi-LSTM and 40 for Spatial RNN. For the word meaning level, we set hidden size as 5 for the RNN, and set the top k as 5 for k-Max pooling, the learning rate is set to 0.001. On the answer selection task, we set the hidden dimension as 50 for Bi-LSTM and 10 for Spatial RNN. For the word meaning level, we set hidden size as 1 for the RNN, and set the topk as 10 for k-Max pooling, the learning rate is set to 0.0001. To train the model, we leverage Adam as our optimizer to update the parameters, and minimize the categorical cross entropy of the training set until the model convergence.

Evaluation Metrics. The PI is the binary classification problem, we adopt accuracy to evaluation the performance on this task. ASS can be considered as a ranking problem, we utilize mean average precision (MAP) and mean reciprocal rank (MRR).

where N is the number of testing ranking lists, M is the number of positive sentence in a ranking list. $S_j^{+(i)}$ is the j-th positive sentence in the i-th ranking list, $r(\cdot)$ denotes the rank of a sentence in the ranking list.

Table 1. Performance on Quora question dataset.

Models	Accuracy (%)
Siamese CNN	79.60
Multi-perspective-CNN	81.38
Siamese-LSTM	82.58
Multi-perspective-LSTM	83.21
L.D.C.	85.55
BiMPM-w/o-tricks	85.88
BiMPM-Full	**88.17**
RI-Match-WL	83.86
RI-Match-WL-w/o-cos	83.42
RI-Match-PSL	82.76
RI-Match-Full	**85.91**

Table 2. Performance on the WikiQA dataset.

Models	MAP	MRR
Word Count	0.652	0.665
ABCNN	0.692	0.711
Attention-CNN	0.689	0.696
Attention-LSTM	0.688	0.707
L.D.C.	0.705	0.723
GRU	0.659	0.669
BiMPM	**0.718**	**0.731**
RI-Match-Full	0.689	0.692

4.2 Paraphrasing Identification

Paraphrasing identification aims to determine whether two sentences tell the same story. In this Sub-section, we compare our model with relatively new baselines on the paraphrasing identification task.

Dataset. To evaluate the effectiveness of our model, we perform our experiments on the dataset of "Quora Question Pairs". To be a fair comparison, we adopt the splitting ways of [15]. This dataset consists of over 400,000 question pairs, and each question pair is annotated with a binary value indicating whether the two questions pairs are paraphrase of each other. The authors randomly select 5,000 paraphrases and 5,000 non-paraphrases as the dev set, and sample another 5,000 paraphrases and 5,000 non-paraphrases as the test set. Then they keep the remaining pairs as the training set. For getting the detail of dataset, please refer to [15][2].

Baseline. To make a sufficient comparison, we choose six relatively new baselines: Siamese CNN [6], Multi-Perspective CNN [2], Siamese-LSTM [10], Multi-Perspective-LSTM [2], L.D.C [16], BiMPM. For these baselines, they all adopt the above-mentioned splitting ways for the datasets.

- For the representation based methods, the Siamese CNN and Siamese-LSTM encode the sentence into dense vectors separately by CNN and LSTM. With the multi-perspective technique, the Siamese models are promoted as Multi-Perspective-CNN and Multi-Perspective-LSTM.
- For the interaction based methods, the L.D.C takes both the similarities and dissimilarities into account by decomposing and composing lexical semantics over texts. And the BiMPM guides the interaction with the attention-base neural architecture by encodes each sentence both in word embedding and

[2] We can obtain the source codes and dataset partition at: https://zhiguowang.github.io.

character embedding level. Both models have obtained the state-of-the art baselines in this datasets.

Performance Comparison. For a fair comparison with above baselines, we directly generate the result under the same setting from the literature. To be fair, we implement the BiMPM without the ticks (*e.g. character embedding, dropout, etc.*), we also list the baseline of BiMPM-Full with these tricks in Table 1. In order to show the influence of different level signals for semantic matching. We test the performance of RI-Match for it contains the signals just in the word level (RI-Match-WL) or the sentence representations level (RI-Match-PSL) and the full model (RI-Match-Full). The results are listed in Table 1. From the results, we could conclude the experimental findings as follows.

1. The models belong to interaction focused methods outperform the representation focused methods. This mainly emphasizes the importance of interaction structure. Representing the sentence to a vector directly will lost many information, this is an important factor that more deep models focus on describing the interaction of the texts.
2. Our model are better than all baselines, achieving the state-of-the-art performance. This illustrates the effectiveness of our model.
3. The RI-Match-Full is better than the RI-Match-WL and RI-Match-PSL, which shows that the signals both in word level and sentence representations level are important for semantic matching. We define more ways to construct the signals in word level rather than the sentence representations level, including word meaning signals and word level interaction signals. Therefore, we can see that RI-Match-WL is 1.1 point percentage higher than the RI-Match-PSL. This gap is obvious in this dataset. For further explanation for this, we check the performance just in word level without the cosine signals (RI-Match-WL-without-cos). We can see that RI-Match-WL is better than RI-Match-WL-without-cos. This means that we can improve performance of model in some extent by defining more different signals in each level.

4.3 Answer Selection

Answer selection is a task to rank the candidate answers based on their matching degree to the question. Evaluation metrics of this tasks are mean average precision (MAP) and mean reciprocal rank (MRR). We experiment on the WikiQA dataset.

Dataset. WikiQA is a public benchmark datasets, we need to rank the candidate answers according to a question. It includes 20,360 question-answer pairs in training set, 1,126 pairs in development set and 2,341 pairs in test set. We filter the questions without the correct answers.

Baseline. To make a fair comparison, we select following baselines.

- Word Count: is non-neural architecture, it calculates the frequency of non-stop words between question and answer.

- GRU: is used to obtain the sentence representations signals, it calculate the similarity of the sentence vector.
- ABCNN, BiMPM: ABCNN and BiMPM belong to the interaction focused methods introduced in related work, they are two state-of-art baselines for this task.
- Attention-based models: both the Attention-based-CNN [17] and Attention-based-LSTM [13] build the attention matrix after sentence representation, they adopt CNN and LSTM separately to encode sentences.

Performance Comparison. For a effective comparison, we report the results under the same setting from the literature. In the Table 2, we can see that RI-Match can do well on this task. Compared with the non-neural architecture, our models can automatically extract more semantic signals from the data, then leading the better performance. For BiMPM, they obtained their best performance by using the character embedding, it can richly capture the information in the input layer. The GRU belongs to the representation focused methods which encode the sentence by GRU. Both the L.D.C and BiMPM are belongs to interaction focused methods, they describe the interaction on the sentence representations level. From above, we can see that deep models basically beat the traditional method. In deep models, describing the interaction of texts sometimes can achieve better result. Our model can also do well in this task except the paraphrasing identification task.

5 Conclusions

In this paper, we propose a deep model by integrating both interactions and representations for deep semantic matching, namely RI-Match. We define various measure functions in each level to produce signals. Moreover, we adopt the spatial RNN to capture the recursive matching structure, which has good performance in many semantic matching tasks. Our model has the good performance in paraphrasing identification and answer selection tasks.

Our models can be further extended in many respects. In the word interactions level based on the word identities, we just adopt the *xnor* to capture the interaction of two words. We can define the more interpretable signals to enrich the signals for the word level interaction. In the future we plan to increase this signals, then the RI-Match will be rich for the semantic matching.

Acknowledgments. This work was funded by the 973 Program of China under Grant No. 2014CB340401, the National Natural Science Foundation of China (NSFC) under Grants No. 61425016, 61472401, 61722211, 61773362, and 20180290, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2016102, and the National Key R&D Program of China under Grants No. 2016QY02D0405.

References

1. Graves, A., Schmidhuber, J.: Offline handwriting recognition with multidimensional recurrent neural networks. In: Advances in Neural Information Processing Systems, pp. 545–552 (2009)
2. He, H., Gimpel, K., Lin, J.: Multi-perspective sentence similarity modeling with convolutional neural networks. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1576–1586 (2015)
3. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Advances in Neural Information Processing Systems, pp. 2042–2050 (2014)
4. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
5. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188 (2014)
6. Leal-Taixé, L., Canton-Ferrer, C., Schindler, K.: Learning by tracking: siamese CNN for robust target association. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 33–40 (2016)
7. Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., Cheng, X.: Text matching as image recognition. In: AAAI, pp. 2793–2799 (2016)
8. Qiu, X., Huang, X.: Convolutional neural tensor network architecture for community-based question answering. In: IJCAI, pp. 1305–1311 (2015)
9. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems, pp. 926–934 (2013)
10. Varior, R.R., Shuai, B., Lu, J., Xu, D., Wang, G.: A siamese long short-term memory architecture for human re-identification. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 135–153. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_9
11. Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., Cheng, X.: A deep architecture for semantic matching with multiple positional sentence representations. In: AAAI, vol. 16, pp. 2835–2841 (2016)
12. Wan, S., Lan, Y., Xu, J., Guo, J., Pang, L., Cheng, X.: Match-SRNN: modeling the recursive matching structure with spatial RNN. arXiv preprint arXiv:1604.04378 (2016)
13. Wang, Y., Huang, M., Zhao, L., et al.: Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 606–615 (2016)
14. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. arXiv preprint arXiv:1702.03814 (2017)
15. Wang, Z., Ittycheriah, A.: FAQ-based question answering via word alignment. arXiv preprint arXiv:1507.02628 (2015)
16. Wang, Z., Mi, H., Ittycheriah, A.: Sentence similarity learning by lexical decomposition and composition. arXiv preprint arXiv:1602.07019 (2016)
17. Yin, W., Schütze, H., Xiang, B., Zhou, B.: ABCNN: attention-based convolutional neural network for modeling sentence pairs. arXiv preprint arXiv:1512.05193 (2015)