Convolutional Interaction Network for Natural Language Inference

Jingjing Gong[‡], **Xipeng Qiu**[*‡], **Xinchi Chen**[‡], **Dong Liang**[§ †], **Xuanjing Huang**[‡]
[‡] Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
[‡] School of Computer Science, Fudan University
[§] State Key Laboratory of Information Security, Institute of Information Engineering CAS
{jjgong15, xpqiu, xinchichen13, xjhuang}@fudan.edu.cn,liangdong@iie.ac.cn

Abstract

Attention-based neural models have achieved great success in natural language inference (NLI). In this paper, we propose the Convolutional Interaction Network (CIN), a general model to capture the interaction between two sentences, which can be an alternative to the attention mechanism for NLI. Specifically, CIN encodes one sentence with the filters dynamically generated based on another sentence. Since the filters may be designed to have various numbers and sizes, CIN can capture more complicated interaction patterns. Experiments on three very large datasets demonstrate CIN's efficacy.

1 Introduction

Natural language inference (NLI) is a pivotal and challenging natural language processing (NLP) task. The goal of NLI is to identify the logical relationship (entailment, neutral, or contradiction) between a premise and a corresponding hypothesis. Generally, NLI is also related to many other NLP tasks under the paradigm of semantic matching of two texts, such as question answering Hu et al. (2014); Wan et al. (2016) and information retrieval Liu et al. (2015), and so on. An essential challenge is to capture the semantic relevance of two sentences. Due to the semantic gap (or lexical chasm) problem, natural language inference is still a challenging problem.

Recently, deep learning is raising a substantial interest in natural language inference and has achieved some great progresses Hu et al. (2014); Parikh et al. (2016); Chen et al. (2017a). To model the complicated semantic relationship between two sentences, previous models heavily utilize various attention mechanism Bahdanau et al. (2014); Vaswani et al. (2017) to build the interaction at different granularity (word, phrase and sentence level), such as ABCNN Yin et al. (2016), Attention LSTM Rocktäschel et al. (2015), bi-directional attention LSTM Chen et al. (2017a), and so on. While attention is very successful in natural language inference, its mechanism is quite simple and can be regarded as a weighted sum of the target vectors. This paradigm results in a lack of flexibility in more complicated interaction model.

In this paper, we propose a new interaction module, called Convolutional Interaction Network (CIN), which can serve as an alternative module of attention mechanism. Specifically, CIN utilizes convolutional neural network to extract the valued features (or representations) from sentences. In the case of NLI, whether a feature of one sentence being important depends on another sentence. Inspired by the idea of using one network to generate the parameters of another network Ha et al. (2016a); De Brabandere et al. (2016), we introduce a filter generation network to dynamically generate convolutional filters. Each sentence is convolved by a dynamically generated filter by another sentence. Thus, the convolved features of one sentence can be regarded as context-aware representations under the influence of another sentence.

The contributions of this paper can be summarized as follows.

1. CIN is a new interaction model, invented as an alternative module to the attention model. CIN can also capture both the intra- or inter-interactions of two sentences.

2. Compared to attention model, CIN is more general and flexible to capture the complicated interaction. As discussed in Section

[*] Corresponding Author.
[†] Contribution during internship at Fudan University.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1576–1585
Brussels, Belgium, October 31 - November 4, 2018.

3.3, the attention model is approximately equivalent to a special case of CIN.

3. We perform extensive empirical studies on three very large datasets. Experiment results demonstrate that our proposed architecture is effective for natural language inference.

2 Attentive Interaction for Natural Language Inference

Currently, the dominative method for natural language inference is to use attention mechanism to model the interaction between two sentence.

Given two input sentences $x = [x_1, x_2, \cdots, x_m]$ and $y = [y_1, y_2, \cdots, y_n]$ with length m and n respectively, we first encode them into two vectorial sequences

$$X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m] \in \mathbb{R}^{d \times m}, \quad (1)$$

$$Y = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n] \in \mathbb{R}^{d \times n}. \quad (2)$$

The encoder usually consists of one or several CNN/RNN layers to get the context-aware token representations.

To capture the interaction between two sentence, various neural attentions can be used, such as sentence2word attention Rocktäschel et al. (2015), word2word attention Parikh et al. (2016); Chen et al. (2017a).

Word2word Attentive Interaction The word2word attention captures the dependency between two words x_i and y_j from the concerned two sentences respectively. The word2word attention computes a similarity matrix M, in which each element $m_{i,j}$ is the alignment score between $\mathbf{x}_i$ and $\mathbf{y}_j$.

$$m_{i,j} = f(\mathbf{x}_i, \mathbf{y}_j), 1 \leq i \leq m, 1 \leq j \leq n, \quad (3)$$

where f is a score function.

There are two most prevalent attention functions: multiplicative attention and additive attention. Multiplicative attention is:

$$f(\mathbf{x}_i, \mathbf{y}_j) = \mathbf{x}_i^{\mathrm{T}} \mathbf{y}_j. \quad (4)$$

Additive attention computes a compatibility function by a feed-forward network with a single hidden layer.

$$f(\mathbf{x}_i, \mathbf{y}_j) = \mathbf{w}^{\top} \sigma(\mathbf{W}_1 \mathbf{x}_i + \mathbf{W}_2 \mathbf{y}_j + \mathbf{b}), \quad (5)$$

where $\mathbf{w}$, $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{b}$ are learnable parameters.

While these two kinds of attentions have similar performance, the multiplicative attention is more popular in practice since it requires less computation power and have less memory demand with optimized matrix multiplication. With multiplicative attention, we can compute the *mimic* representations for both X and Y.

$$\bar{X} = Y \mathrm{softmax}(Y^{\mathrm{T}} X) \in \mathbb{R}^{d \times m}, \quad (6)$$

$$\bar{Y} = X \mathrm{softmax}(X^{\mathrm{T}} Y) \in \mathbb{R}^{d \times n}, \quad (7)$$

where $\mathrm{softmax}(\cdot)$ is column-wise normalization function. Each vector $\bar{\mathbf{x}}_i \in \bar{X}$ is called as *mimic vector*, which is a weighted summation of $\{\mathbf{y}_j\}_{j=1}^n$. Intuitively, the mimic vector $\bar{\mathbf{x}}_i$ provides the related information of token x_i extracted from sentence Y.

Prediction After interaction, a prediction module is used to aggregate the interaction information and extract the fix-length representation of two sentences. Finally, the final sentence representations are fed into a feed-forward network to predict the relationship between two sentences.

3 Convolutional Interaction Network

In this section, we propose a new interaction method by utilizing dynamic convolutional filters, called Convolutional Interaction Network (CIN). CIN can serve as an alternative module of attention mechanism.

We first briefly introduce how the convolution works over text sequence, then describe the proposed model and its connection to attention model.

3.1 Convolution over Sequence

Convolution is an effective operation in deep neural networks, which convolves the input with a set of filters to extract non-linear compositional features. Although originally designed for computer vision, convolutional models have subsequently shown to be effective for NLP and have achieved excellent performance in sentence modeling Kim (2014); Kalchbrenner et al. (2014), and other traditional NLP tasks Hu et al. (2014); Zeng et al. (2014); Gehring et al. (2017).

Given a sentence representation $X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m] \in \mathbb{R}^{d \times m}$, a convolutional filter $W^{(f)} \in \mathbb{R}^{d \times kd}$, the convolution process is defined as

$$\mathbf{x}'_i = f\left(W^{(f)}[\mathbf{x}_{i-[k/2]}, \cdots, \mathbf{x}_{i+[k/2]}] + \mathbf{b}^{(f)}\right), \quad (8)$$

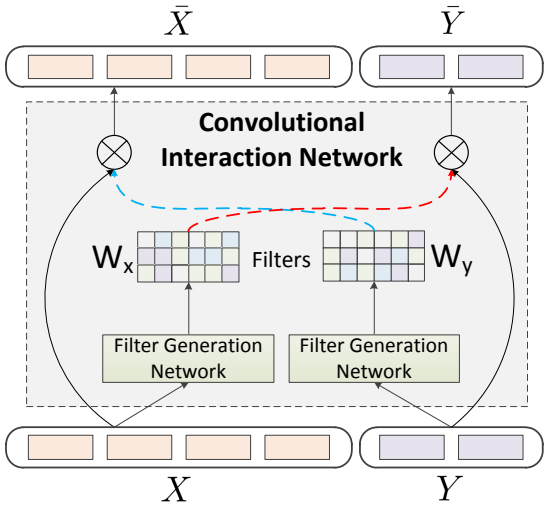

Figure 1: Convolutional Interaction Network. $\otimes$ denotes the convolution operation.

where $f(\cdot)$ is a non-linear activation function, such as ReLU, k indicates the size of convolution window, and $\mathbf{b}^{(f)} \in \mathbb{R}^d$ is a bias vector.

The convolution can be abbreviated as

$$X' = f(W^{(f)} \otimes X) \in \mathbb{R}^{d \times m}, \tag{9}$$

where $\otimes$ denotes the convolutional operation. To ensure the output of convolution has equal length as to the input, we pad $[\frac{k}{2}]$ zero vectors on both sides of the input.

3.2 Convolutional Interaction Network

Convolution is very effective when it comes to extracting useful features from a sentence. But for NLI, whether a word (or feature) being important in one sentence depends on another sentence. Therefore, a better convolution operation should have the ability to extract substantial features from one sentence according to another sentence. Thus, the convolutional filter should be dynamically changeable. Inspired by Jia et al. (2016); Ha et al. (2016b), we propose a *filter generation network* (FGN) to generate a dynamical filter, which is used to extract the context-aware information.

Given two sentences x, y, and their representations $X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m] \in \mathbb{R}^{d \times m}$ and $Y = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n] \in \mathbb{R}^{d \times n}$, the filter for each sentence is generated according to the other sentence by

$$W_x^{(f)} = \text{FGN}(X) \in \mathbb{R}^{d \times \tau d}, \tag{10}$$

$$W_y^{(f)} = \text{FGN}(Y) \in \mathbb{R}^{d \times \tau d}, \tag{11}$$

where τ is the width of filter, $\text{FGN}(\cdot)$ is the filter generation network. A detailed implementation of FGN is presented in Section 3.4.

Now we can convolve the two sentences with the generated filters.

$$\bar{X} = f(W_y^{(f)} \otimes X) \in \mathbb{R}^{d \times m}, \tag{12}$$

$$\bar{Y} = f(W_x^{(f)} \otimes Y) \in \mathbb{R}^{d \times n}, \tag{13}$$

where the attained matrix $\bar{X}$ and $\bar{Y}$ can be regarded as the context-aware representation of sentences x and y, depending on each other.

Figure 1 gives an illustration of CIN.

3.3 Connection to Attentive Interaction

CIN is more general than attention model. Assuming that we set $k = 1$ and FGN to be a function of $\text{FGN}(X) = XX^{\mathrm{T}}$, Eq. (12) and (13) of CIN can be written as

$$\bar{X} = (YY^{\mathrm{T}})X = Y(Y^{\mathrm{T}}X), \tag{14}$$

$$\bar{Y} = (XX^{\mathrm{T}})Y = X(X^{\mathrm{T}}Y). \tag{15}$$

Compared to Eq. (6) and (7), under the above assumption, CIN is equivalent to the word2word multiplicative attention model without softmax normalization.

3.4 An Implementation of Filter Generation Network (FGN)

To generate the dynamic filters, the key factor is how to choose the filter generation network $\text{FGN}(\cdot)$ in Eq. (10) and (11). Although many sophisticated networks can be employed, we give an simple implementation in this paper.

For ease of presentation, we only describe how we generate dynamical filter according to sentence x. The same procedure is utilized for sentence y.

Firstly, we summarize the information of sentence x with an over-time k-max pooling on X,

$$U_x = \text{ReLU}(W_u \otimes X) \in \mathbb{R}^{d \times m}, \tag{16}$$

$$\mathbf{z}_x^{1:k} = \text{k-max}(U_x) \in \mathbb{R}^{d \times k}, \tag{17}$$

where U_x is a non-linear transformation of X by convolution filter $W_u \in \mathbb{R}^{d \times d}$. The idea of k-max pooling is to capture the most important features (the k highest values) from sentence X.

Then we generate k filters W_x^j for $j = 1, \cdots, k$ by

$$W_x^j = \text{ReLU}\left(P\text{diag}(\mathbf{z}_x^j)Q + B\right), \tag{18}$$

where $P \in \mathbb{R}^{\frac{d}{k} \times d}$, $Q \in \mathbb{R}^{d \times \tau d}$ and $B \in \mathbb{R}^{\frac{d}{k} \times \tau d}$ are learnable parameters.

The final filter is obtained by concatenating the k generated filters,

$$W_x^{(f)} = [W_x^1; W_x^2; \cdots; W_x^k] \in \mathbb{R}^{d \times \tau d}. \quad (19)$$

Similar to x, we can also obtain the dynamic filters $W_y^{(f)}$ according to the sentence y.

4 Incorporating CIN into a Deep Network Architecture for NLI

Our overall network architecture for NLI is based on a successful model proposed by Chen et al. (2017a). The major difference is that we use CIN to capture the interaction, instead of bi-directional attention.

The network architecture consists of three components: (1) an encoding layer; (2) convolutional interaction layers; (3) a prediction layer. Figure 2 gives an illustration.

4.1 Encoding Layer

The input of natural language inference task is a pair of sentences x and y. Since each word in a sentence is a symbol that can not be directly processed by neural networks, we need first map each word to a d dimensional embedding vector.

Thus, the two sentences are mapped to two matrix $E_x \in \mathbb{R}^{d_e \times m}$ and $E_y \in \mathbb{R}^{d_e \times n}$ respectively. We also use syntactical and lexical information such as part of speech tagging information, exact match feature and character representation. In this paper, exact match value of each word is set to 1(default to be 0) if the word concerned share the same stem or lemma with any word in counterpart sentence. Character representation of the word is obtained using a convolution neural network followed by a max pooling along sequence length dimension as same as Kim (2014). The final representation of word is a concatenation of word embedding, character encoded vector, POS tagging embedding and exact match feature. Both character embedding and POS tagging embedding are randomly initialized. All embeddings are updated during training.

We use bi-directional LSTM (BiLSTM) Hochreiter and Schmidhuber (1997) to incorporate the forward and backward context information of sequence. Thus, we can get the

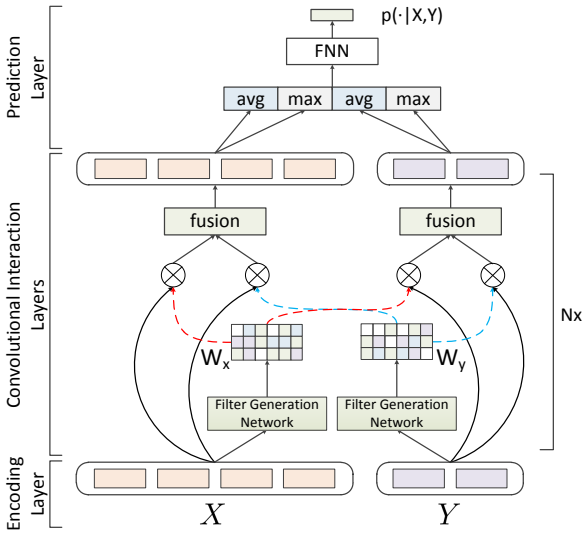

Figure 2: The overall network architecture for natural language inference. The N_x means the number of the stacking interaction layers.

phrase-level encoding of two input sentences,

$$X = [\text{Bi-LSTM}(E_x); E_x], \quad (20)$$
$$Y = [\text{Bi-LSTM}(E_y); E_y], \quad (21)$$

where $X \in \mathbb{R}^{d \times m}$ and $Y \in \mathbb{R}^{d \times n}$ are the phrase-level encoding representation of sentence x and y respectively.

4.2 Convolutional Interaction Layers

In the interaction layers, we use our proposed CIN to model the interaction between two sentences.

We first dynamically generate context-aware filters $W_x^{(f)}$ and $W_y^{(f)}$ based on the sentence encodings X and Y respectively, which are further used for both intra-sentence and inter-sentence interaction.

Intra-Sentence Interaction The intra-sentence convolutional interaction is to convolve one sentence by the filter generated by itself.

$$X_{intra} = f(W_x^{(f)} \otimes X), \quad (22)$$
$$Y_{intra} = f(W_y^{(f)} \otimes Y), \quad (23)$$

The role of the intra-sentence convolutional interaction is the same as self-attention Shen et al. (2017), which is also very useful in NLI.

Inter-Sentence Interaction The inter-sentence interaction takes filters generated from the coun-

terpart sentence to convolve the inputs.

$$X_{inter} = f(W_x^{(f)} \otimes Y), \quad (24)$$

$$Y_{inter} = f(W_y^{(f)} \otimes X), \quad (25)$$

The inter-sentence convolutional interaction plays a role similar to the cross-attention between two sentences.

Fusion Layer After CIN, we can fuse two kinds of context-aware representations of each sentence. For sentence x, the X_{intra} and X_{inter} represent the extracted features under consideration of information of itself and sentence y respectively.

To efficiently utilize X_{intra} and X_{inter}, a fusion layer is used. We use the comparing operation proposed in Chen et al. (2017a) to fuse the two kinds of representation. Let $\mathbf{u}_i$ and $\mathbf{v}_i$ be intra and inter attentive vector of the i-th word in sentence x, a heuristic and effective composition operator is used to combine two vectors.

$$\bar{\mathbf{x}}_i^{(c)} = [\mathbf{u}_i; \mathbf{v}_i; \mathbf{u}_i - \mathbf{v}_i; \mathbf{u}_i \odot \mathbf{v}_i; |\mathbf{u}_i - \mathbf{v}_i|], \quad (26)$$

$$\mathbf{x}_i^{(c)} = \text{ReLU}(\mathbf{W}_c \bar{\mathbf{x}}_i^c + \mathbf{W}_x \mathbf{x}_i + \mathbf{b}_c), \quad (27)$$

Thus, we can obtain two fused representations $X^{(c)}$ and $Y^{(c)}$ for two sentences, which are further fed into the prediction layer or another stacked interaction layer. The interaction layers can be stacked for N_x times to capture the complicated matching information.

4.3 Prediction Layer

After interaction layers, an aggregation layer is employed to aggregate the two sequences of fustion vectors $X^{(c)}$ and $Y^{(c)}$ into a fixed-length matching vectors. The aggregation component usually consists of another BiLSTM layer and a following pooling layer. We then perform max pooling over time for both $X^{(c)}$ and $Y^{(c)}$ to get two fix representation vector for two sentences, $\mathbf{p}$ and $\mathbf{h}$:

$$\mathbf{p} = \max(X^{(c)}), \quad (28)$$

$$\mathbf{h} = \max(Y^{(c)}), \quad (29)$$

where the functions $\max$ is the max pooling operations over time steps.

Finally, the pooled vectors are composed as one relation vector and fed into a feed-forward network to predict the relationship between two sentences. Specially, the two-layer feed-forward network has one hidden layers with $tanh$ activation and $softmax$ output layer in our experiments.

$$\mathbf{m} = [\mathbf{p}; \mathbf{h}; \mathbf{p} - \mathbf{h}; \mathbf{p} * \mathbf{h}; |\mathbf{p} - \mathbf{h}|], \quad (30)$$

$$p(\cdot|x, y) = \text{FNN}(\mathbf{m}). \quad (31)$$

	Train	Dev	Test	Len(P)	Len(H)	Vocab
SNLI	549K	9.8K	9.8K	14	8	36K
MultiNLI[1]	392K	9.8K	9.8K	22	11	85K
MultiNLI[2]	392K	9.8K	9.8K	22	11	85K
Quora	384K	10K	10K	12	12	107K

Table 1: Statistics of three datasets: SNLI, MultiNLI, Quora. Len(P) and Len(H) refer to the average length of two sentences. MultiNLI[1] and MultiNLI[2] indicate the in-domain and cross-domain datasets.

5 Training

Given a trainset $\{x^{(i)}, y^{(i)}, t^{(i)}\}_{i=1}^N$, the objective is to minimize a cross entropy loss $\mathcal{J}(\theta)$:

$$\mathcal{J}(\theta) = -\frac{1}{N}\sum_i \log p(t^{(i)}|x^{(i)}, y^{(i)}) + \lambda||\theta||_2^2, \quad (32)$$

where θ represents all the connection weights.

We use the Adam optimizer Kingma and Ba (2014) with an initial learning rate of 0.0004. Default L2 regularization λ is set to 10^{-6}. To avoid overfitting, dropout is applied after each fully connected, recurrent or convolutional layer.

Initialization We take advantage of pre-trained word embeddings such as Glove Pennington et al. (2014) to transfer more knowledge from vast unlabeled data. For the words that don't appear in Glove, we randomly initialize their embeddings from a normal distribution with mean 0.0 and standard deviation 0.1.

The network weights are initialized with Xavier normalization Glorot and Bengio (2010) to maintain the variance of activations throughout the forward and backward passes. Biases are uniformly set to zero when the network is constructed.

5.1 Datasets

To make quantitative evaluation, our model was evaluated on three well known datasets: Stanford Natural Language Inference dataset (SNLI), MultiNLI dataset and Quora Question pair dataset (Quora). Detailed statistical information of these datasets is shown in Table 1.

Models	Train	Test
Handcrafted features (Bowman et al., 2015)	99.7	78.2
LSTM with attention (Rocktäschel et al., 2015)	85.3	83.5
Match-LSTM (Wang and Jiang, 2016)	92.0	86.1
Decomposable attention model (Parikh et al., 2016)	90.5	86.8
BiMPM (Zhiguo Wang, 2017)	90.9	87.5
NTI-SLSTM-LSTM (Munkhdalai and Yu, 2017)	88.5	87.3
Re-read LSTM (Sha et al., 2016)	90.7	87.5
DIIN (Gong et al., 2017)	91.2	88.0
ESIM (Chen et al., 2017a)	92.6	88.0
CIN	**93.2**	**88.0**
ESIM (Chen et al., 2017a) (Ensemble)	93.5	88.6
BiMPM (Zhiguo Wang, 2017) (Ensemble)	93.2	88.8
DIIN (Gong et al., 2017) (Ensemble)	92.3	88.9
CIN (Ensemble)	**94.3**	**89.1**

Table 2: Performance on SNLI dataset.

SNLI The SNLI corpus Bowman et al. (2015) consists of 570,152 sentence pairs. Each sentence pair is labeled as one of entailment, contradiction and neutral relation.

MultiNLI Orgnized the same as SNLI, MultiNLI corpus Williams et al. (2017) is another dataset for NLI, it contains 433,000 sentence pairs. Like SNLI, each pair is labeled with one of entailment, contradiction and neutral label. Difference between MultiNLI and SNLI is that, MultiNLI have in-domain test set and development set as well as an out-of-domain test and development set.

Quora The Quora Question pair dataset have over 400k question pairs, each question pair is assigned with a binary label to indicate if the pair are paraphrase to each other. We evaluate our model on the data which was previously partitioned by Zhiguo Wang (2017)

5.2 Overall Results

We use the accuracy to evaluate the performance of our convolutional interaction network (CIN) and other models on SNLI, MultiNLI and Quora.

SNLI Table 2 shows the results of different models on the train set and test set of SNLI. The first row gives a baseline model with handcrafted features presented by Bowman et al. (2015). All the other models are attention-based neural networks. Wang and Jiang (2016) exploits the long short-term memory (LSTM) for NLI. Parikh et al. (2016) uses attention to decompose the problem into subproblems that can be solved separately. Chen et al. (2017a) incorporates the chain LSTM and tree LSTM jointly. Zhiguo Wang (2017) proposes a bilateral multi-perspective matching for NLI.

In Table 2, the second block gives the single models. As we can see, our proposed model CIN achieves 88.0% in accuracy on SNLI test set. Compared to the previous work, CIN obtains competitive performance.

To further improve the performance of NLI systems, researchers have built ensemble models. Ensemble systems obtained the best performance on SNLI. Our ensemble model obtains 89.1% in accuracy and outperforms the current state-of-the-art model.

Overall, single model of CIN performs competitively well and outperforms the previous models on ensemble scenarios for the natural language inference task.

MultiNLI Table 3 shows the performance of different models on MultiNLI. The original aim of this dataset is to evaluate the quality of sentence representations. Recently this dataset is also used to evaluate the interaction model involving attention mechanism.

The first line of Table 3 gives a baseline model without interaction. The second block of Table 3 gives the attention-based models. The proposed

Models	Match	Mismatch
BiLSTM (Williams et al., 2017)	67.0	67.6
InnerAtt (Balazs et al., 2017)	72.1	72.1
ESIM (Chen et al., 2017a)	72.3	72.1
Gated-Att BiLSTM (Chen et al., 2017b)	73.2	73.6
ESIM (Chen et al., 2017a)	76.3	75.8
CIN	**77.0**	**77.6**

Table 3: Performance on MultiNLI test set.

Models	Test
Siamese-CNN	79.60
Multi-Perspective CNN	81.38
Siamese-LSTM	82.58
Multi-Perspective-LSTM	83.21
L.D.C	85.55
BiMPM (Zhiguo Wang, 2017)	88.17
CIN	**88.62**

Table 4: Performance on Quora question pair dataset.

model, CIN, achieves the accuracies of 77.0% and 77.6% on the *match* and *mismatch* test sets respectively. The results show that our model outperforms the other models.

Quora Table 4 shows the performance of different models on the Quora test set. The baselines on Table 4 are all implemented in Zhiguo Wang (2017). The Siamese-CNN model and Siamese-LSTM model encode sentences with CNN and LSTM respectively, and then predict the relationship between them based on the cosine similarity. Multi-Perspective-CNN and Multi-Perspective-LSTM are transformed from Siamese-CNN and Siamese-LSTM respectively by replacing the cosine similarity calculation layer with their multi-perspective cosine matching function. The L.D.C is a general compare-aggregate framework that performs word-level matching followed by a aggregation of convolution neural networks. As we can see, our model outperforms the baselines and achieve 88.62% in the test sets of Quora corpus.

Models	Dev	Test
CIN	**88.6**	**88.0**
Remove whole interaction	85.6	85.1
Remove intra-attention	88.1	87.7

Table 5: Ablation experiment on SNLI dataset.

Premise
(1) A girl playing a violin along with a group of people
(2) A girl playing a violin along with a group of people
Hypothesis
(1) A girl is playing an instrument .
(2) A girl is playing an instrument .

Table 6: Gradient visualization of premise and hypothesis. (1) Gradient scale of X, Y on encoding layer. (2) Gradient scale of $X^{(c)}$, $Y^{(c)}$ on first CIN layer. Darker color corresponds to a higher scale of gradient, and implies a higher contribution to the final prediction.

5.3 Model Ablation

To better understand the performance of our model, we analyze the effect of each key component of the proposed model. As illustrated in Table 5, the first row is the full CIN model. By dropping convolutional interaction layers, the performance decreases to 85.1% on the test set, which indicate the interaction information is crucial for NLI. By just dropping intra-attention layer, the performance decreases to 87.7% on the test set. According to the results, all of the components positively contribute to the final performance.

5.4 Case Study

To give an intuitive understanding of how our model works, we give an analysis on the following case from the test set.

Premise: A girl playing a violin along with a group of people.
Hypothesis: A girl is playing an instrument.
Label: Entailment.

The visualization results are produced from model with two stacked CINs. X, Y is the hidden states at encoding layer, and $X^{(c)}$, $Y^{(c)}$ is the hidden states at first CIN layer. For a hidden state

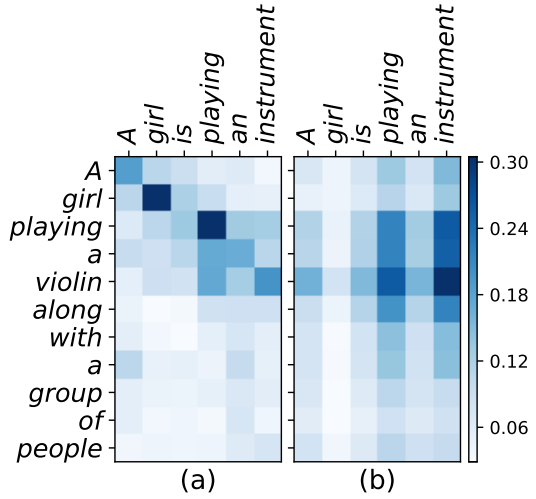

Figure 3: A visualization of word to word correlation. Darker color correspond to a higher correlation. (a) Correlation of $X^{\mathrm{T}}Y$ at encoder layer. (b) Correlation of $X^{(c)^{\mathrm{T}}}Y^{(c)}$ at first CIN layer.

$\mathbf{x}_i$ of word x_i, we can calculate its gradient scale $||\frac{\partial \mathcal{J}}{\partial \mathbf{x}_i}||^2$ to show its contribution to final prediction.

Table 6 shows the gradient scales of hidden states of each word in the encoding layer and the first CIN layer. As we can see, some phrases (like *playing a violin* and *playing an instrument*) instead of isolated words (like *violin* and *instrument*) become more focused after a CIN layer. It implies CIN could capture some higher level patterns.

Figure 3 gives a visualization of correlations of hidden states of two sentences. (a) shows the correlations after the encoding layer, the same words are most correlated. This is because embedding layer and encoding layer are shared between premise and hypothesis. (b) shows the correlations after the first CIN layer, the correlation exists between phrases {*playing a violin* vs. *playing an instrument*}, instead of the same words. The interaction layer connects *playing* in Premise to Hypothesis *instrument*, and connects *playing* in Hypothesis to Premise *violin*. Thus, the correlation between *instrument* in Hypothesis and *violin* in Premise are boosted, as we know these are important to reasoning.

6 Related Work

There are mainly two threads of work related to ours.

One thread of work is using attention-based model for natural language inference (NLI). NLI has been widely investigated for many years. Benefiting from the development of deep learning and the availability of large-scale annotated datasets, deep neural models have achieved great success. Rocktäschel et al. (2015) firstly use LSTM with attention for text matching task. Wang and Jiang (2016) use word-by-word attention to exploit the word-level match. Parikh et al. (2016) propose a new framework to model the relationship between two sentences using interact-compare-aggregate architecture. Chen et al. (2017a) incorporates the chain LSTM and tree LSTM jointly. Zhiguo Wang (2017) use self-attention mechanism to capture contextual information from the whole sentence.

Unlike the above models, we use an alternative model to capture the complicate interaction information of two sentences.

Another thread of work is the idea of using one network to generate the parameters of another network. De Brabandere et al. (2016) proposed the dynamic filter network to implicitly learn a variety of filtering operations. Ha et al. (2016a) proposed the model hypernetwork, which uses a small network to generate the weights for a larger network.

Unlike these models, our dynamical filter is employed for interaction. Therefore, a filter generation function is proposed to capture the related intra and inter dependent information of a sentence pair.

7 Conclusion

In this paper, we propose an alternative interaction model, Convolutional Interaction Network (CIN), for natural language inference. CIN utilizes the dynamic convolutional filters to model the interaction between two sentences. Specifically, each sentence is convolved by dynamical filters generated based on another sentence. CIN is more general and flexible since the filters may have various numbers and sizes, thereby capturing more complicated interaction patterns. Experiments on three very large datasets demonstrate the efficacy of our proposed model.

In future work, we hope to improve the extensibility of CIN and apply it to other NLP tasks, such as machine comprehension.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. We would like to thank the anonymous reviewers for their valuable comments. The research work is

supported by Shanghai Municipal Science and Technology Commission (No. 17JC1404100 and 16JC1420401), and National Natural Science Foundation of China (No. 61672162 and 61751201), Fundamental Theory and Cutting-Edge Technology Research Program of Institute of Information Engineering, CAS (Grant No. Y7Z0281102).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Jorge A Balazs, Edison Marrese-Taylor, Pablo Loyola, and Yutaka Matsuo. 2017. Refining raw sentence representations for textual entailment recognition via attention. *arXiv preprint arXiv:1707.03103*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017a. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1657–1668.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Recurrent neural network-based sentence encoder with gated attention for natural language inference. *arXiv preprint arXiv:1708.01353*.

Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. 2016. Dynamic filter networks. In *Neural Information Processing Systems (NIPS)*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1243–1252.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256.

Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*.

David Ha, Andrew Dai, and Quoc V Le. 2016a. Hypernetworks. *arXiv preprint arXiv:1609.09106*.

David Ha, Andrew Dai, and Quoc V Le. 2016b. Hypernetworks. *arXiv preprint arXiv:1609.09106*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in Neural Information Processing Systems*.

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. 2016. Dynamic filter networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 667–675.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *NAACL*.

Tsendsuren Munkhdalai and Hong Yu. 2017. Neural tree indexers for text understanding. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 1, page 11. NIH Public Access.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.

Lei Sha, Baobao Chang, Zhifang Sui, and Sujian Li. 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*, pages 2870–2879.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2017. Disan: Directional self-attention network for rnn/cnn-free language understanding. *arXiv preprint arXiv:1709.04696*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.

Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *AAAI*.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with lstm. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: attention-based convolutional neural network for modeling sentence pairs. *TACL*, 4:259–272.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344.

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