Abstract

Natural language inference aims to predict whether a premise sentence can infer another hypothesis sentence. Existing methods typically have framed the reasoning problem as a semantic matching task. The both sentences are encoded and interacted symmetrically and in parallel. However, in the process of reasoning, the role of the two sentences is obviously different, and the sentence pairs for NLI are asymmetrical corpora. In this paper, we propose an asynchronous deep interaction network (ADIN) to complete the task. ADIN is a neural network structure stacked with multiple inference sub-layers, and each sub-layer consists of two local inference modules in an asymmetrical manner. Different from previous methods, this model deconstructs the reasoning process and implements the asynchronous and multi-step reasoning. Experiment results show that ADIN achieves competitive performance and outperforms strong baselines on three popular benchmarks: SNLI, MultiNLI, and SciTail.

1 Introduction

Natural language inference (NLI) is a pivotal and fundamental task in natural language understanding and artificial intelligence. The goal of NLI is to predict whether a premise sentence can infer another hypothesis sentence. As illustrated in Table 1, logical relationships between the two sentences include entailment (if the premise is true, then the hypothesis must be true), contradiction (if the premise is true, then the hypothesis must be false), and neutral (neither entailment nor contradiction).

As a core task, conventional approaches have studied various aspects of the inference problem (MacCartney and Manning, 2008; Heilman and Smith, 2010). Thanks to the release of the largest publicly available corpus - the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015), neural network-based models have also been successfully used for this task (Parikh et al., 2016; Chen et al., 2016; Tay et al., 2018; Duan et al., 2018). These methods typically treat the premise sentence and the hypothesis sentence equally, learn an alignment of sub-phrases in both sentences symmetrically and in parallel, and fuse local information for making a global decision at the sentence level. They all frame the inference problem as a semantic matching task and ignore the reasoning process.

However, different from a simple semantic matching task, reasoning should be asynchronous and fully interpretable (Yi et al., 2018). Moreover, the sentence pairs for NLI are asymmetrical corpora, i.e., \( I(a,b) \neq I(b,a) \). Considering the first
example in Table 1, the premise sentence can infer
the hypothesis sentence, however, the hypothesis
sentence can’t infer the premise sentence. The in-
ference process intuitively needs to consider the
relationship between two sentences in sequential
order. According to the actual inference process,
we argue that the model should first get the in-
ferential information to model the hypothesis sen-
tence, based on the premise sentence, and then
model premise sentence, based on the new repre-
sentation for hypothesis sentence.

In this paper, we propose an asynchronous deep
interaction network (ADIN) to achieve the reason-
ing. This model is stacked with multiple inference
sub-layers to implement the multi-step reasoning,
and each sub-layer consists of two local inference
modules in an asymmetrical manner to simulate
the asynchronous and interpretable reasoning pro-
cess. In a local inference module, we update the
sentence representation by using the local infer-
ence information, based on the attention of the
other sentence. Lastly, we combine the inference
information between the two sentences to make a
global decision.

To demonstrate the effectiveness of our model,
we evaluate it on three popular benchmarks:
SNLI, MultiNLI, and SciTail. The experimen-
tal results on these three data sets reveal that our
method achieves competitive performance.

The main contributions of this work can be sum-
marized as follows:

- We break the matching architecture that inter-

acts with the information between two sen-
tences for alignment, and propose an asyn-
chronous deep interaction network to achieve
the asynchronous and multi-step reasoning.

- We deconstruct the reasoning process be-
 tween the two sentences, and the process can
be analyzed step-by-step.

- The experimental results on three highly
competitive benchmark datasets demonstrate
that our model can achieve better perfor-
mance than other strong baselines.

2 Approach

We define the natural language inference as a
classification task that predicts the relation $y \in
Y$ for a given pair of sentences, where $Y = \{entailment, contradiction, neutral\}$. In this
work, we propose an asynchronous deep interac-
tion network (ADIN) to complete this task. The
overall architecture of the model is illustrated on
the left part of Figure 1.

Our sentence inference architecture, ADIN , is
composed of the following three components: (1) infor-
mation representation layer converts the two
sentences into semantic representations; (2) asyn-
chronous inference layer produces new representa-
tions for the two sentences, based on the infer-
ence information; and (3) interaction and predic-
tion layer determines the overall inference rela-
tionship between a premise and hypothesis.
2.1 Local Inference Module

Given two natural sentences $a$ and $b$, $H_a = \{h_{a_i}, h_{a_i} \in \mathbb{R}^k, i = 1, 2, ..., m\}$ and $H_b = \{h_{b_j}, h_{b_j} \in \mathbb{R}^k, j = 1, 2, ..., n\}$ denote their $k$-dimensional representations respectively, where $m, n$ denote the length of two sentences. Here, we implement a general reasoning process where the module captures the relevance between the two sentences, then incorporates the inferential information to the new representation for sentence $b$, based on the sentence $a$. First, we compute a co-attention matrix $E \in \mathbb{R}^{m \times n}$ to capture the relevance between the two sentences, each element $E_{i,j} \in R$ indicates the relevance between the i-th word of sentence $a$ and the j-th word of $b$. Formally, the co-attention matrix could be computed as:

$$E_{i,j} = P^T \tanh(W(h_{a_i} \circ h_{b_j})), \quad (1)$$

where $W \in \mathbb{R}^{s \times k}$, $P \in \mathbb{R}^s$, and $\circ$ denotes the element-wise production operation. Then, we get $a$-guided attentive vectors for sentence $b$:

$$e_{b_j} = \text{softmax}(E_{:,j}), \quad (2)$$

$$h_{b_j}' = H_a \cdot e_{b_j}, \quad (3)$$

In order to enhance the interaction further, we combine the original vector and $a$-guided attentive vector for sentence $b$. More formally:

$$h_{b_j}'' = [h_{b_j}; h_{b_j}' \cdot h_{b_j}; h_{b_j} \circ h_{b_j}], \quad (4)$$

$$\tilde{h}_{b_j} = \text{ReLU}(W_{b_j} h_{b_j}'' + b_{b_j}), \quad (5)$$

where $[\cdot; \cdot; \cdot; \cdot]$ refers to the concatenation operation. In Equation 4, we first calculate the difference and the element-wise product for $(h_{b_j}, h_{b_j}')$. We get the new representation containing $a$-guided inferential information for sentence $b$:

$$\tilde{H}_b = (\tilde{h}_{b_1}, \tilde{h}_{b_2}, ..., \tilde{h}_{b_n}), \quad (6)$$

$$\hat{H}_b = \text{LayerNorm}(\tilde{H}_b), \quad (7)$$

Where $\text{LayerNorm}(\cdot)$ is layer normalization (Ba et al., 2016). The result $\hat{H}_b$ is a 2D-tensor that has the same shape as $H_b$, and we refer to the whole inferential module as:

$$\text{InferentialModule}(H_a, \hat{H}_b), \quad (8)$$

As described, the inferential module can capture the relevance between the two sentences, incorporate the inferential information to the new representation for sentence $b$, based on the sentence $a$.

2.2 Information Representation Layer

The information representation layer converts each word or phrase in the sentences into a vector representation and constructs the representation matrix for the sentences. We combine the multi-level features as the sentence representation. Each token is represented as a vector by using the pre-trained word embedding such as GloVe (Pennington et al., 2014), word2Vec (Mikolov et al., 2013), and fasttext (Joulin et al., 2016). It can also incorporate more syntactical and lexical information into the feature vector.

For ADIN, we use a concatenation of word embedding, character embedding, and syntactical features as the sentence representation. The word embedding is obtained by mapping token to high dimensional vector space by pre-trained word vector (300D Glove 840B), and the word embedding is updated during training. Character-level embedding could alleviate out-of-vocabulary (OOV) problems and capture helpful morphological information. As in (Kim et al., 2016; Lee et al., 2016), we filter the character embedding with 1D convolution kernel. The character convolutional feature maps are then max pooled over the time dimension for each token to obtain a vector.

As in (Chen et al., 2018), the syntactical features consist of one-hot part-of-speech (POS) tagging feature and binary exact match (EM) feature. For one sentence, the EM value is activated if the same word is found in the other sentence.

Next, ADIN adopts bidirectional Long Short-Term Memory network (Bi-LSTM) (Graves and Schmidhuber, 2005) to model the internal temporal interaction on both directions of the sentences. Consider a premise sentence $p$ and a hypothesis sentence $q$, we have got their multi-level features representation. Suppose the length of $p$ and $q$ are $m$ and $n$ respectively. These multi-level features representation are then passed to a Bi-LSTM encoder to obtain the context-dependent hidden state matrix, i.e., $H_p = \{h_{p_i}, h_{p_i} \in \mathbb{R}^d, i = 1, 2, ..., m\}$, and $H_q = \{h_{q_j}, h_{q_j} \in \mathbb{R}^d, j = 1, 2, ..., n\}$, where $d$ is the dimension of Bi-LSTM’s hidden state.

2.3 Asynchronous Inference Layer

Recently, along with the development of deep learning methods, some neural attention-based models have also been successfully used for NLI (Rocktäschel et al., 2015; Parikh et al., 2016; Duan et al., 2018). However, these methods typically
frame the inference problem as a semantic matching task and ignore the reasoning process, where the premise sentence and the hypothesis sentence are encoded and interacted symmetrically and in parallel.

In this paper, we utilize the local inference module to deconstruct the reasoning process and achieve the asynchronous and multi-step reasoning for NLI. To model the multi-step reasoning habit, this model is stacked with \( N \) inference sub-layers to capture step-by-step the logic relationship between the two sentences. In the each inference sub-layer, two inferential modules perform two asynchronous inference processes respectively.

Concretely, in the \( t \)-th inference sub-layer, given the representations of two sentences computed in the previous sub-layer: \( V_{p}^{t-1} = (v_{p_1}^{t-1}, v_{p_2}^{t-1}, ..., v_{p_m}^{t-1}) \) and \( V_{q}^{t-1} = (v_{q_1}^{t-1}, v_{q_2}^{t-1}, ..., v_{q_n}^{t-1}) \), we get the deeper-level representations:

\[
V_{q}^{t} = \text{InferentialModule}(V_{p}^{t-1}, V_{q}^{t-1}), \quad (9)
\]

\[
V_{p}^{t} = \text{InferentialModule}(V_{q}^{t-1}, V_{p}^{t-1}), \quad (10)
\]

\[
\hat{v}_{q_i} = \text{Bi-LSTM}^{t}(v_{q_{i-1}}^{t}, \hat{v}_{q_{i+1}}^{t}), \quad (11)
\]

\[
\hat{v}_{p_j} = \text{Bi-LSTM}^{t}(v_{p_{j-1}}^{t}, \hat{v}_{p_{j+1}}^{t}), \quad (12)
\]

where \( V_{p}^{0} = H_{p}, V_{q}^{0} = H_{q}, \hat{V}_{p}^{t} = (\hat{v}_{p_1}^{t}, \hat{v}_{p_2}^{t}, ..., \hat{v}_{p_m}^{t}) \), and \( \hat{V}_{q}^{t} = (\hat{v}_{q_1}^{t}, \hat{v}_{q_2}^{t}, ..., \hat{v}_{q_n}^{t}) \). In an inference sub-layer, we first get the inferential information to update the representation for hypothesis sentence, based on the premise sentence. Next, the model incorporates the inferential information to the premise sentence, based on the new representation for hypothesis sentence.

### 2.4 Interaction and Prediction Layer

To extract a proper representation for each sentence, we apply a mean pooling and a max pooling on each of them. Formally:

\[
V_{p}^{\text{mean}} = \frac{1}{m} \sum_{i=1}^{m} v_{p_i}^{N}, \quad V_{p}^{\text{max}} = \max_{i=1}^{m} v_{p_i}^{N}, \quad (14)
\]

\[
V_{q}^{\text{mean}} = \frac{1}{n} \sum_{j=1}^{n} v_{q_j}^{N}, \quad V_{q}^{\text{max}} = \max_{j=1}^{n} v_{q_j}^{N}, \quad (15)
\]

\[
V_{p}^{\text{new}} = [V_{p}^{\text{mean}}, V_{p}^{\text{max}}], \quad V_{q}^{\text{new}} = [V_{q}^{\text{mean}}, V_{q}^{\text{max}}], \quad (16)
\]

Then, we aggregate these representations \( V_{p}^{\text{new}} \) and \( V_{q}^{\text{new}} \) for the two sentences \( p \) and \( q \) in various ways in the interaction layer and the final feature vector \( r \) for the inference is obtained as follows:

\[
r = [V_{p}^{\text{new}}; V_{q}^{\text{new}}; V_{p}^{\text{new}} - V_{q}^{\text{new}}; V_{p}^{\text{new}} \odot V_{q}^{\text{new}}], \quad (17)
\]

Finally, based on the aggregated feature \( r \), we use a multi-layer perceptron (MLP) classifier to predict the label:

\[
v = \text{ReLU}(W_r r + b_r), \quad (18)
\]

\[
\hat{y} = \text{softmax}(W_y v + b_y). \quad (19)
\]

where \( W_r, b_r, W_y, \) and \( b_y \) are trainable parameters. The entire model is trained end-to-end, optimizing the standard multi-class cross-entropy loss function.

### 3 Experiments

In this section, we present the evaluation of our model. We first perform quantitative evaluation, comparing our model with other strong baselines. We then conduct some qualitative analyses to understand how ADIN achieve the asynchronous and multi-step inference between the premise sentence and the hypothesis sentence.

#### 3.1 Dataset

We evaluate our model on three popular benchmarks: the Stanford Natural Language Inference (SNLI), the MultiGenre NLI Corpus (MultiNLI) and SciTail. Detailed statistical information of these datasets is shown in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
<th>Len(P)</th>
<th>Len(H)</th>
<th>Vocab</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNLI</td>
<td>549K</td>
<td>9.8K</td>
<td>9.8K</td>
<td>14</td>
<td>8</td>
<td>36K</td>
</tr>
<tr>
<td>MultiNLI(^1)</td>
<td>9.8K</td>
<td>9.8K</td>
<td>22</td>
<td>11</td>
<td>85K</td>
<td></td>
</tr>
<tr>
<td>MultiNLI(^2)</td>
<td>392K</td>
<td>9.8K</td>
<td>9.8K</td>
<td>22</td>
<td>11</td>
<td>85K</td>
</tr>
<tr>
<td>SciTail</td>
<td>23.6K</td>
<td>1.3K</td>
<td>2.1K</td>
<td>10</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Statistics of datasets: SNLI, MultiNLI, SciTail. 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.
**MultiNLI** The corpus (Williams et al., 2017) is a new dataset for NLI, which contains 433k sentences pairs. Similar to SNLI, each pair is labeled with one of the following relationships: entailment, contradiction, or neutral. We compare on two test sets (matched and mismatched) which represent in-domain and out-domain performance. We use the same data split as provided by (Williams et al., 2017).

**SciTail** We also include the newly released SciTail dataset (Khot et al., 2018) which is a binary entailment classification task constructed from science questions. This is the first entailment set that is created solely from natural sentences that already exist independently “in the wild” rather than sentences authored specifically for the entailment task. We use the same data split as in (Khot et al., 2018).

### 3.2 Models for Comparing

To analyze the effectiveness of our model, we evaluate some traditional and state-of-the-art methods as baselines as follows on the above three data sets:

- **DecompAtt** (Parikh et al., 2016) is a simple model that decomposes the problem into parallelizable attention computations.

- **ESIM** (Chen et al., 2016) is a previous state-of-the-art model for the natural language inference (NLI) task. It is a sequential model that incorporates the chain LSTM and the tree LSTM to infer local information between two sentences.

- **BiMPM** is proposed in (Wang et al., 2017). The model combines two sentence encoders and employs a multi-perspective matching mechanism in sentence pair modeling tasks.

- **DIIN** (Gong et al., 2017) is a novel class of neural network architectures that is able to achieve high-level understanding of the sentence pair by hierarchically extracting semantic features from the interaction space. The model uses word-by-word dimension-wise alignment tensors to encode the high-order alignment relationship between sentence pairs.

- **DGEM** (Khot et al., 2018) is a entailment model that exploits structure from the hypothesis only. This model shows the value of structured representation on just the hypothesis for NLI.

- **MwAN** (Tan et al., 2018) is a multiway attention network that applies multiple attention functions to model the matching between a pair of sentences.

- **CAFE** (Tay et al., 2018) compares and compresses alignment pairs using factorization layers which leverages the rich history of standard machine learning literature to achieve this task.

- **AF-DMN** (Duan et al., 2018) stacks multiple computational blocks in its matching layer to learn the interaction of the sentence pair better.

- **KIM** (Chen et al., 2018) is neural network-based NLI model that can benefit from external knowledge. The model is capable of leveraging external knowledge in co-attention, local inference collection, and inference composition components.

### 3.3 Experiment Configurations

Hyper-parameters may influence the performance of a neural network-based model. For all the three datasets, there are 3 inference sub-layers in the asynchronous inference Layer. An Adam (Kingma and Ba, 2014) optimizer with $\beta_1$ as 0.9 and $\beta_2$ as 0.999 is used to optimize all trainable parameters. The initial learning rate is set to 0.001 and is halved when the accuracy on the dev set decreases. We also apply dropout (Srivastava et al., 2014) on all MLPs to avoid over-fitting, and the dropout rate is set to 0.2. For preprocessing, we just tokenize the sentences and lowercase the tokens.

For initialization, we initialize the word embeddings with a 300D Glove 840B (Pennington et al., 2014), and the out-of-vocabulary (OOV) words are randomly initialized. All word embeddings are updated during training. Parameters, including neural network parameters and OOV word embeddings, are initialized with a uniform distribution between $[-0.01, 0.01]$. The character embeddings are randomly initialized with 100D. We crop or pad each token to have 16 characters. And the 1D convolution kernel size for character embedding is 5.
3.4 Ensemble

The ensemble strategy is an effective method to improve model accuracy. Following (Wang et al., 2017), our ensemble model averages the probability distributions from five individual single ADINs, who have exactly identical architectures but distinguished initializations on parameters.

3.5 Quantitative Results

We use the accuracy to evaluate the performance of ADIN and other models on datasets SNLI, MultiNLI, and SciTail.

Table 3 shows the results of different models on the training and test sets of SNLI. In Table 3, the first category of methods are single models and the second category of methods are ensemble models. We show our model, ADIN, achieves state-of-the-art performance on the competitive leaderboard. In this table, KIM is neural network-based NLI model that can benefit from external knowledge, and other strong baselines encode and interact the both sentences symmetrically and in parallel.

Table 4 reports our results on the MultiNLI dataset. Similar to Table 3, the first category of methods are single models and the second category of methods are ensemble models. On MultiNLI, we compare on two test sets (matched and mismatched) which represent in-domain and out-domain performance. ADIN significantly outperforms ESIM, a strong baseline on the both test sets. An ensemble of ADIN models also achieve competitive result on the MultiNLI dataset.

As illustrated in Table 5, our model outperforms the baselines and achieves an accuracy of 84.6% in the test set of the SciTail dataset. As such, empirical results demonstrate the effectiveness of our proposed ADIN model on the challenging SciTail dataset.

For the results on all three datasets, we conduct the students paired t-test. For SNLI and MultiNLI, the p-value of the significance test between the results of our model and AF-DMN is less than 0.01 and 0.05, respectively. For SciTail, the p-value of the significance test between the results of our model and CAFE is also less than 0.01. These results further prove the effectiveness of our model.
Table 5: Comparison with previous models on the SciTail dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESIM (Chen et al., 2016)</td>
<td>70.6</td>
</tr>
<tr>
<td>DecompAtt (Parikh et al., 2016)</td>
<td>72.3</td>
</tr>
<tr>
<td>DGEM (Khot et al., 2018)</td>
<td>77.3</td>
</tr>
<tr>
<td>CAFE (Tay et al., 2018)</td>
<td>83.3</td>
</tr>
<tr>
<td>ADIN (ours)</td>
<td><strong>84.6</strong></td>
</tr>
</tbody>
</table>

Table 6: Effect of number of asynchronous inference layers on the SNLI.

<table>
<thead>
<tr>
<th># layers</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88.6</td>
<td>88.4</td>
</tr>
<tr>
<td>2</td>
<td>88.9</td>
<td>88.6</td>
</tr>
<tr>
<td>3</td>
<td>89.0</td>
<td>88.8</td>
</tr>
</tbody>
</table>

3.6 Model Analysis

To better understand the performance of ADIN, we analyze the effect of each key component of the proposed model on the SNLI dataset. Table 6 shows the performance with a different number of asynchronous inference sub-layers. As we can see, with the number of sub-layers increases from 1 to 3, the performance increases both on the development set and the test set. As the level of reasoning deepens, the model captures more inferential information. Because of computational cost, we just set the number of sub-layers as 3 on SNLI and other two datasets.

In Table 7, we show the results of ablation study on our base model. After removing the Bi-LSTM in the asynchronous inference Layer, the model performance decrease by 0.3 percentage points on the test set. Furthermore, we study the effect of two inferential modules in one asynchronous inference sub-layer. Without the first inferential module, that is, without the reasoning process from premise to hypothesis, the model performance sharply decreases by 0.8 percentage points. However, remove the second module and the test accuracy decreases by 0.5 percentage points. (exchanged inference order) indicates that we get the inferential information to first model the premise sentence, and then model hypothesis sentence. The performance of the model is reduced to 88.3% after exchanging inference order between two sentences. The above three experiments reflect that the both modules are not equally important for the inference and the sentence pair for NLI is asymmetrical corpora. In the last comparative experiment, we explore the role of multi-level features. We remove character embedding and syntactical features and just keep word embedding as the representation. The test accuracy is reduced to 88.2%.

3.7 Case study

To visually demonstrate the validity of the model, we do a qualitative study using the first example in Table 1.

<table>
<thead>
<tr>
<th>Models</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADIN (-Bi-LSTM)</td>
<td>88.7</td>
<td>88.5</td>
</tr>
<tr>
<td>ADIN (-first inferential module)</td>
<td>88.4</td>
<td>88.0</td>
</tr>
<tr>
<td>ADIN (-second inferential module)</td>
<td>88.5</td>
<td>88.3</td>
</tr>
<tr>
<td>ADIN (exchanged inference order)</td>
<td>88.6</td>
<td>88.3</td>
</tr>
<tr>
<td>ADIN (-char-emb - syntactical fea)</td>
<td>88.5</td>
<td>88.2</td>
</tr>
<tr>
<td>ADIN (ours)</td>
<td><strong>89.0</strong></td>
<td><strong>88.8</strong></td>
</tr>
</tbody>
</table>

Table 7: Effect of components on the SNLI.

Figure 2: Gradient visualization of premise and hypothesis. (1) Gradient scale of $H_p, H_q$ on representation layer. (2) Gradient scale of $V^1_p, V^1_q$ on the first asynchronous inference sub-layer. (3) Gradient scale of $V^2_p, V^2_q$ on the second asynchronous inference sub-layer. (4) Gradient scale of $V^3_p, V^3_q$ on the third asynchronous inference sub-layer. Darker color corresponds to a higher scale of gradient, and implies a higher contribution to the final prediction.

an animal is playing with a plastic toy.
(2) An animal is playing with a plastic toy.
(3) An animal is playing with a plastic toy.
(4) An animal is playing with a plastic toy.

Figure 2 gives a visualization of the contribution to the final prediction of every word. As we can see, some phrases (like jumping for a Frisbee and play-
ing with a plastic toy) instead of isolated words (like Frisbee and toy) become more focused after an asynchronous inference layer. The results imply that ADIN could capture some higher-level patterns. As the level of reasoning deepens, the model captures more inferential information.

4 Related Work

As a long standing problem in NLP research, natural language inference (or textual entailment recognition) has been widely investigated for many years. Conventional works on NLI relies on handcrafted features such as syntactic information, n-gram overlapping and so on (Bowman et al., 2015; Heilman and Smith, 2010).

Benefiting from the development of deep learning and the availability of large-scale annotated datasets (Bowman et al., 2015), neural network-based models have also been successfully used for this task. And two categories of neural network-based models have been developed for this problem. The first set of models is sentence encoding-based and aims to find vector representation for each sentence and classifies the relation by using the concatenation of two vector representation (Bowman et al., 2016; Nangia et al., 2017; Mou et al., 2015). However, this kind of framework ignores the interaction between two sentences.

The other set of models uses the cross-sentence feature or inter-sentence attention from one sentence to another, and is hence referred to as a matching-aggregation framework. Parikh et al. (2016) use attention to decompose the problem into subproblems that can be solved separately, thus making it trivially parallelizable. Chen et al. (2016) propose a state-of-the-art model for the natural language inference (NLI) task. It is a sequential model that incorporates the chain LSTM and the tree LSTM to infer local information between two sentences. A novel class of neural network architectures is proposed in (Gong et al., 2017) that is able to achieve high-level understanding of the sentence pair by hierarchically extracting semantic features from interaction space. Tan et al. (2018) propose a multiway attention network that designs four attention functions to match words in corresponding sentences, aggregates the matching information from each function, and combines the information from all functions to obtain the final representation. Tay et al. (2018) compare and compress alignment pairs using factorization layers which leverages the rich history of standard machine learning literature to achieve this task. AF-DMN (Duan et al., 2018) stacks multiple computational blocks in its matching layer to learn the interaction of the sentence pair better. KIM (Chen et al., 2018) is capable of leveraging external knowledge in co-attention, local inference collection, and inference composition components to improve the performance. These methods all frame the inference problem as a semantic matching task and ignore the reasoning process.

Different from the above methods, ADIN is a neural network structure stacked with multiple asynchronous inference sub-layers, and each sub-layer consists of two local inference modules in an asymmetrical manner. This model deconstructs the reasoning process and implements the asynchronous and multi-step reasoning.

5 Conclusions

In this paper, we propose an asynchronous deep interaction network (ADIN) for natural language inference. To simulate human reasoning process, ADIN is stacked with multiple asynchronous inference sub-layers, and each sub-layer consists of two inferential modules in an asymmetrical manner. The model deconstructs the reasoning process and implements the asynchronous and multi-step reasoning. We evaluate our model on three popular benchmarks: SNLI, MultiNLI, and SciTail. The experiment results show that ADIN achieves competitive performance and outperforms strong baselines.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by China National Key RD Program (No. 2018YFB1005104, 2018YFC0831105, 2017YFB1002104), National Natural Science Foundation of China (No. 61976056, 61532011, 61751201), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), STCSM (No.16JC1420401,17JC1420200).

References


Michael Heilman and Noah A Smith. 2010. Tree edit models for recognizing textual entailments, phrases, and answers to questions. *ACL*.


