Asynchronous Deep Interaction Network for Natural Language Inference

Di Liang[*], Fubao Zhang[*], Qi Zhang[†], and Xuanjing Huang
School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing
Fudan University, Shanghai, P.R.China 201203
{liangd17, fbzhang17, qz, xjhuang}@fudan.edu.cn

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.

Premise: A dog is jumping for a Frisbee in the snow. **Hypothesis:** An animal is playing with a plastic toy. **Label:** Entailment
Premise: He was crying like his mother had just walloped him. **Hypothesis:** He was crying like his mother hit him with a spoon. **Label:** Neutral
Premise: Several men in front of a white building. **Hypothesis:**Several people in front of a gray building. **Label:** contradiction

Table 1: Examples of natural language inference.

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

[*]Equal contribution. Alphabetical order of the last name.
[†]Corresponding author.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing, pages 2692–2700,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

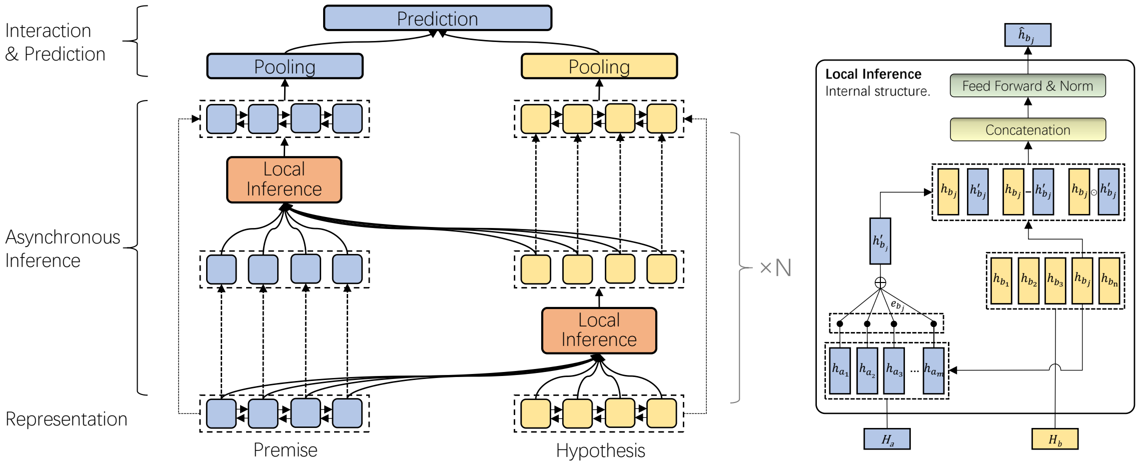


Figure 1: The overall view of our model. The left part is the main framework of this work. The dashed lines refer to the copy operation. The asynchronous inference layer is stacked with N inference sub-layers. In the first sub-layer of the asynchronous inference layer, the input is from the representation layer. Subsequently, the input of sub-layers come from the previous sub-layers. The right part is the detailed structure of the local inference module, taking $\hat{\mathbf{h}}_{b_j}$ as an example.

example in Table 1, the premise sentence can infer the hypothesis sentence, however, the hypothesis sentence can't infer the premise sentence. The inference 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 inferential information to model the hypothesis sentence, based on the premise sentence, and then model premise sentence, based on the new representation for hypothesis sentence.

In this paper, we propose an asynchronous deep interaction network (ADIN) to achieve the reasoning. 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 process. In a local inference module, we update the sentence representation by using the local inference 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 experimental results on these three data sets reveal that our method achieves competitive performance.

The main contributions of this work can be summarized as follows:

- We break the matching architecture that interacts with the information between two sentences for alignment, and propose an asynchronous deep interaction network to achieve the asynchronous and multi-step reasoning.

- We deconstruct the reasoning process between 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 performance 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 interaction 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) *information representation layer* converts the two sentences into semantic representations; (2) *asynchronous inference layer* produces new representations for the two sentences, based on the inference information; and (3) *interaction and prediction layer* determines the overall inference relationship between a premise and hypothesis.

2.1 Local Inference Module

Given two natural sentences a and b, $\mathbf{H}_a = \{\mathbf{h}_{a_i} | \mathbf{h}_{a_i} \in \mathbb{R}^k, i = 1, 2, ..., m\}$ and $\mathbf{H}_b = \{\mathbf{h}_{b_j} | \mathbf{h}_{b_j} \in \mathbb{R}^k, j = 1, 2, ..., n\}$ 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 $\mathbf{E}_{i,j} \in R$ 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:

$$\mathbf{E}_{i,j} = \mathbf{P}^T tanh(\mathbf{W}(\mathbf{h}_{a_i} \odot \mathbf{h}_{b_j})), \tag{1}$$

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

$$\mathbf{e}_{b_j} = softmax(\mathbf{E}_{:,j}), \tag{2}$$

$$\mathbf{h}'_{b_j} = \mathbf{H}_a \cdot \mathbf{e}_{b_j}, \tag{3}$$

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

$$\mathbf{h}''_{b_j} = [\mathbf{h}_{b_j}; \mathbf{h}'_{b_j}; \mathbf{h}_{b_j} - \mathbf{h}'_{b_j}; \mathbf{h}_{b_j} \odot \mathbf{h}'_{b_j}], \tag{4}$$

$$\tilde{\mathbf{h}}_{b_j} = ReLU(\mathbf{W}_{b_j} \mathbf{h}''_{b_j} + \mathbf{b}_{b_j}), \tag{5}$$

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

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

$$\hat{\mathbf{H}}_b = LayerNorm(\tilde{\mathbf{H}}_b), \tag{7}$$

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

$$\textbf{InferentialModule}(\mathbf{H}_a, \mathbf{H}_b), \tag{8}$$

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, $\mathbf{H}_p = \{\mathbf{h}_{p_i} | \mathbf{h}_{p_i} \in \mathbb{R}^d, i = 1, 2, ..., m\}$, and $\mathbf{H}_q = \{\mathbf{h}_{q_j} | \mathbf{h}_{q_j} \in \mathbb{R}^d, j = 1, 2, ..., n\}$, 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 : $\mathbf{V}_p^{t-1} = (\mathbf{v}_{p_1}^{t-1}, \mathbf{v}_{p_2}^{t-1}, ..., \mathbf{v}_{p_m}^{t-1})$ and $\mathbf{V}_q^{t-1} = (\mathbf{v}_{q_1}^{t-1}, \mathbf{v}_{q_2}^{t-1}, ..., \mathbf{v}_{q_n}^{t-1})$, we get the deeper-level representations:

$$\hat{\mathbf{V}}_q^t = \textbf{InferentialModule}(\mathbf{V}_p^{t-1}, \mathbf{V}_q^{t-1}), \tag{9}$$

$$\hat{\mathbf{V}}_p^t = \textbf{InferentialModule}(\mathbf{V}_q^{t}, \mathbf{V}_p^{t-1}), \tag{10}$$

$$\tilde{\mathbf{v}}_{q_i}^t = [\hat{\mathbf{v}}_{q_i}^t; \mathbf{v}_{q_i}^{t-1}], \quad \tilde{\mathbf{v}}_{p_j}^t = [\hat{\mathbf{v}}_{p_j}^t; \mathbf{v}_{p_j}^{t-1}], \tag{11}$$

$$\mathbf{v}_{q_i}^t = \text{Bi-LSTM}(\tilde{\mathbf{v}}_{q_{i-1}}^t, \tilde{\mathbf{v}}_{q_i}^t, \tilde{\mathbf{v}}_{q_{i+1}}^t), \tag{12}$$

$$\mathbf{v}_{p_j}^t = \text{Bi-LSTM}(\tilde{\mathbf{v}}_{p_{j-1}}^t, \tilde{\mathbf{v}}_{p_j}^t, \tilde{\mathbf{v}}_{p_{j+1}}^t), \tag{13}$$

where $\mathbf{V}_p^0 = \mathbf{H}_p$, $\mathbf{V}_q^0 = \mathbf{H}_q$, $\hat{\mathbf{V}}_p^t = (\hat{\mathbf{v}}_{p_1}^t, \hat{\mathbf{v}}_{p_2}^t, ..., \hat{\mathbf{v}}_{p_m}^t)$, and $\hat{\mathbf{V}}_q^t = (\hat{\mathbf{v}}_{q_1}^t, \hat{\mathbf{v}}_{q_2}^t, ..., \hat{\mathbf{v}}_{q_n}^t)$. 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:

$$\mathbf{V}_p^{mean} = \sum_{i=1}^{m} \frac{\mathbf{v}_{p_i}^N}{m}, \quad \mathbf{V}_p^{max} = \max_{i=1}^{m} \mathbf{v}_{p_i}^N, \tag{14}$$

$$\mathbf{V}_q^{mean} = \sum_{j=1}^{n} \frac{\mathbf{v}_{q_j}^N}{n}, \quad \mathbf{V}_q^{max} = \max_{j=1}^{n} \mathbf{v}_{q_j}^N, \tag{15}$$

$$\mathbf{V}_p^{new} = [\mathbf{V}_p^{mean}; \mathbf{V}_p^{max}], \quad \mathbf{V}_q^{new} = [\mathbf{V}_q^{mean}; \mathbf{V}_q^{max}], \tag{16}$$

	Train	Dev	Test	L(P)	L(H)	Vocab
SNLI	549K	9.8K	9.8K	14	8	36K
MultiNLI[1]	392K	9.8K	9.8K	22	11	85K
MultiNLI[2]		9.8K	9.8K	22	11	85K
SciTail	23.6K	1.3K	2.1K	10	7	-

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

Then, we aggregate these representations $\mathbf{V}_p^{new}$ and $\mathbf{V}_q^{new}$ for the two sentences p and q in various ways in the interaction layer and the final feature vector $\mathbf{r}$ for the inference is obtained as follows:

$$\mathbf{r} = [\mathbf{V}_p^{new}; \mathbf{V}_q^{new}; \mathbf{V}_p^{new} - \mathbf{V}_q^{new}; \mathbf{V}_p^{new} \odot \mathbf{V}_q^{new}], \tag{17}$$

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

$$\boldsymbol{v} = ReLU(\mathbf{W}_r\mathbf{r} + \mathbf{b}_r), \tag{18}$$

$$\hat{\boldsymbol{y}} = softmax(\mathbf{W}_v\boldsymbol{v} + \mathbf{b}_v). \tag{19}$$

where $\mathbf{W}_r, \mathbf{b}_r, \mathbf{W}_v$, and $\mathbf{b}_v$ 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.

SNLI is a collection of 570k human written sentence pairs based on image captioning, supporting the task of natural language inference (Bowman et al., 2015). The labels are composed of entailment, neutral and contradiction. The data splits are provided in (Bowman et al., 2015).

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 β_1 as 0.9 and β_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 the 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.

Model	train	test
Single Models		
200D DecompAtt (Parikh et al., 2016)	90.5	86.8
600D ESIM (Chen et al., 2016)	92.6	88.0
BiMPM (Wang et al., 2017)	90.9	87.5
448D DIIN (Gong et al., 2017)	91.2	88.0
300D MwAN (Tan et al., 2018)	94.5	88.3
300D CAFE (Tay et al., 2018)	89.8	88.5
300D AF-DMN (Duan et al., 2018)	94.5	88.6
KIM (Chen et al., 2018)	94.1	88.6
ADIN (ours)	93.6	**88.8**
Ensemble Models		
600D ESIM (Chen et al., 2016)	93.5	88.6
BiMPM (Wang et al., 2017)	93.2	88.8
448D DIIN (Gong et al., 2017)	92.3	88.9
300D AF-DMN (Duan et al., 2018)	94.9	89.0
300D CAFE (Tay et al., 2018)	92.5	89.3
ADIN (ours)	95.6	**89.5**

Table 3: Comparison with previous models on the SNLI dataset.

Model	Test Accuracy	
	Matched	Mismatched
Single Models		
ESIM (Chen et al., 2016)	72.3	72.1
DIIN (Gong et al., 2017)	78.8	77.8
AF-DMN (Duan et al., 2018)	76.9	76.3
CAFE (Tay et al., 2018)	78.7	77.9
MwAN (Tan et al., 2018)	78.5	77.7
ADIN (ours)	**78.8**	**77.9**
Ensemble Models		
DIIN (Gong et al., 2017)	80.0	78.7
CAFE (Tay et al., 2018)	80.2	79.0
MwAN (Tan et al., 2018)	79.8	79.4
ADIN (ours)	**80.3**	**79.6**

Table 4: Comparison with previous models on the MultiNLI dataset.

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.

Model	Accuracy
ESIM (Chen et al., 2016)	70.6
DecompAtt (Parikh et al., 2016)	72.3
DGEM (Khot et al., 2018)	77.3
CAFE (Tay et al., 2018)	83.3
ADIN (ours)	**84.6**

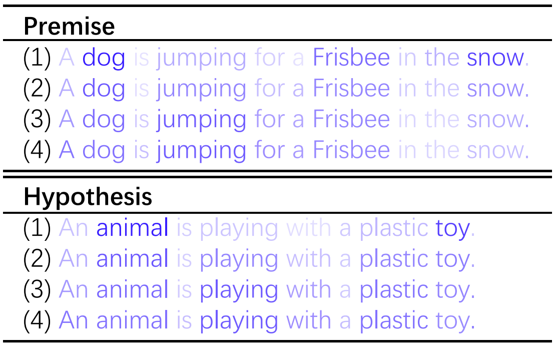
Table 5: Comparison with previous models on the SciTail dataset.

# layers	Dev	Test
1	88.6	88.4
2	88.9	88.6
3	89.0	88.8

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

Models	Dev	Test
(-Bi-LSTM)	88.7	88.5
(-first inferential module)	88.4	88.0
(-second inferential module)	88.5	88.3
(exchanged inference order)	88.6	88.3
(-char-emb - syntactical fea)	88.5	88.2
ADIN (ours)	89.0	**88.8**

Table 7: Effect of components on the SNLI.



Premise
(1) A dog is jumping for a Frisbee in the snow.
(2) A dog is jumping for a Frisbee in the snow.
(3) A dog is jumping for a Frisbee in the snow.
(4) A dog is jumping for a Frisbee in the snow.

Hypothesis
(1) 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: Gradient visualization of premise and hypothesis. (1) Gradient scale of $\mathbf{H}_p, \mathbf{H}_q$ on representation layer. (2) Gradient scale of $\mathbf{V}_p^1, \mathbf{V}_q^1$ on the first asynchronous inference sub-layer. (3) Gradient scale of $\mathbf{V}_p^2, \mathbf{V}_q^2$ on the second asynchronous inference sub-layer. (4) Gradient scale of $\mathbf{V}_p^3, \mathbf{V}_q^3$ 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.

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 dateset.

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 datesets.

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.

$\mathbf{H}_p, \mathbf{H}_q$ are the hidden states at the representation layer of premise sentence and hypothesis sentence, and $\mathbf{V}_p^t, \mathbf{V}_q^t$ are the hidden states at the t-th asynchronous inference sub-layer. For a hidden state $\mathbf{h}_{p_i}$ of word p_i, we can calculate the gradient scale $\left\|\frac{\partial \mathcal{J}}{\partial \mathbf{h}_{p_i}}\right\|^2$ to show its contribution to the final prediction, where $\mathcal{J}$ is the cross-entropy loss. 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

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Samuel R Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. *arXiv preprint arXiv:1603.06021*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2406–2417.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

Chaoqun Duan, Lei Cui, Xinchi Chen, Furu Wei, Conghui Zhu, and Tiejun Zhao. 2018. Attention-fused deep matching network for natural language inference. In *IJCAI*, pages 4033–4040.

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

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *AAAI*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*, pages 2741–2749.

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

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *arXiv preprint arXiv:1610.03017*.

Bill MacCartney and Christopher D Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 521–528. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2015. Natural language inference by tree-based convolution and heuristic matching. *arXiv preprint arXiv:1512.08422*.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel R Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. *arXiv preprint arXiv:1707.08172*.

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 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*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Chuanqi Tan, Furu Wei, Wenhui Wang, Weifeng Lv, and Ming Zhou. 2018. Multiway attention networks for modeling sentence pairs. In *IJCAI*, pages 4411–4417.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. In *Proceedings of EMNLP*, pages 1565–1575.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

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

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *Advances in Neural Information Processing Systems*, pages 1039–1050.

—