Attention-Fused Deep Matching Network for Natural Language Inference

Chaoqun Duan[1*], **Lei Cui**[2], **Xinchi Chen**[3*], **Furu Wei**[2], **Conghui Zhu**[1] and **Tiejun Zhao**[1]
[1] Harbin Institute of Technology, Harbin, China
[2] Microsoft Research Asia, Beijing, China
[3] School of Computer Science, Fudan University, Shanghai, China
cqduan@stu.hit.edu.cn, {lecu, fuwei}@microsoft.com,
xinchichen13@fudan.edu.cn, {conghui, tjzhao}@hit.edu.cn

Abstract

Natural language inference aims to predict whether a premise sentence can infer another hypothesis sentence. Recent progress on this task only relies on a shallow interaction between sentence pairs, which is insufficient for modeling complex relations. In this paper, we present an attention-fused deep matching network (AF-DMN) for natural language inference. Unlike existing models, AF-DMN takes two sentences as input and iteratively learns the attention-aware representations for each side by multi-level interactions. Moreover, we add a self-attention mechanism to fully exploit local context information within each sentence. Experiment results show that AF-DMN achieves state-of-the-art performance and outperforms strong baselines on Stanford natural language inference (SNLI), multi-genre natural language inference (MultiNLI), and Quora duplicate questions datasets.

1 Introduction

Natural language inference (NLI) is a core challenge in the natural language processing community, which aims to predict whether a premise sentence can infer another hypothesis sentence [MacCartney, 2009]. This problem is usually viewed as a classification problem. Existing approaches for NLI are categorized into two types: conventional discrete feature-based approaches [Bowman *et al.*, 2015] and neural network-based models [Chen *et al.*, 2017a].

Neural network-based models have attracted more attention for their ability in assisting efforts in feature engineering [Wang and Jiang, 2015; Chen *et al.*, 2017a; Wang *et al.*, 2017b]. Despite their success, there are still two problems.

The first problem is **long-term context dependency**. Context dependency is essential for NLI, but is difficult to model, especially long-term context dependency. Most previous models [Bowman *et al.*, 2015; Liu *et al.*, 2016b] who focus on modeling the contextual information adopt a long short-term memory network (LSTM) [Hochreiter and Schmidhuber, 1997] or a gated recurrent units network (GRU) [Chung *et al.*, 2014]. However, there exist in NLI some sentences ($L \geq 17$) whose long-term dependency cannot be effectively modeled by LSTM and GRU because of their length.

The second problem is **insufficient model complexity**. Recent evidence [Simonyan and Zisserman, 2014; Szegedy *et al.*, 2015] reveals that network depth is of crucial importance. Previous neural network-based models can be categorized into two classes: sentence encoding based models and attention-based models. For the first class, Bowman *et al.* [2015], Tan *et al.* [2015] propose matching models that use the sentence vectors. This method is simple and effective, but neglects the interaction between two sentences. For the second class, Wang *et al.* [2017b]; Chen *et al.* [2017a] have designed new matching models that leverage attentions from two directions, where bidirectional information is added to achieve higher accuracy. However, the network structures of previous models are still so shallow that the model capability is not sufficient for modeling complex relations.

In order to address these two problems, we propose an attention-fused deep matching network (AF-DMN) for the NLI task. AF-DMN is a neural network structure stacked with multiple computational blocks in its matching layer. Each computational block consists of four sub-layers: (1) a cross attention layer; (2) a fusion layer for cross attention; (3) a self-attention layer; and (4) another fusion layer for self-attention. In this model, we utilize two heterogeneous attention mechanisms in each computational block: cross attention aims to make information interaction between two sentences, while self-attention aims to exploit long-term context dependency. In addition, we employ fusion layers to refine the representation following the two attention layers respectively. Experiments on the SNLI, the MultiNLI and the Quora duplicate questions datasets demonstrate that AF-DMN significantly improves accuracy and achieves state-of-the-art performance.

Our contributions are summarized as follows:

- The AF-DMN model incorporates two attention mechanisms jointly: cross attention aims to make information interaction between two sentences; while self-attention aims to exploit long-term context dependency.
- We first conduct multiple stacked computational blocks in the matching layer for NLI, which allows the model

[*] Contribution during internship at Microsoft Research Asia.

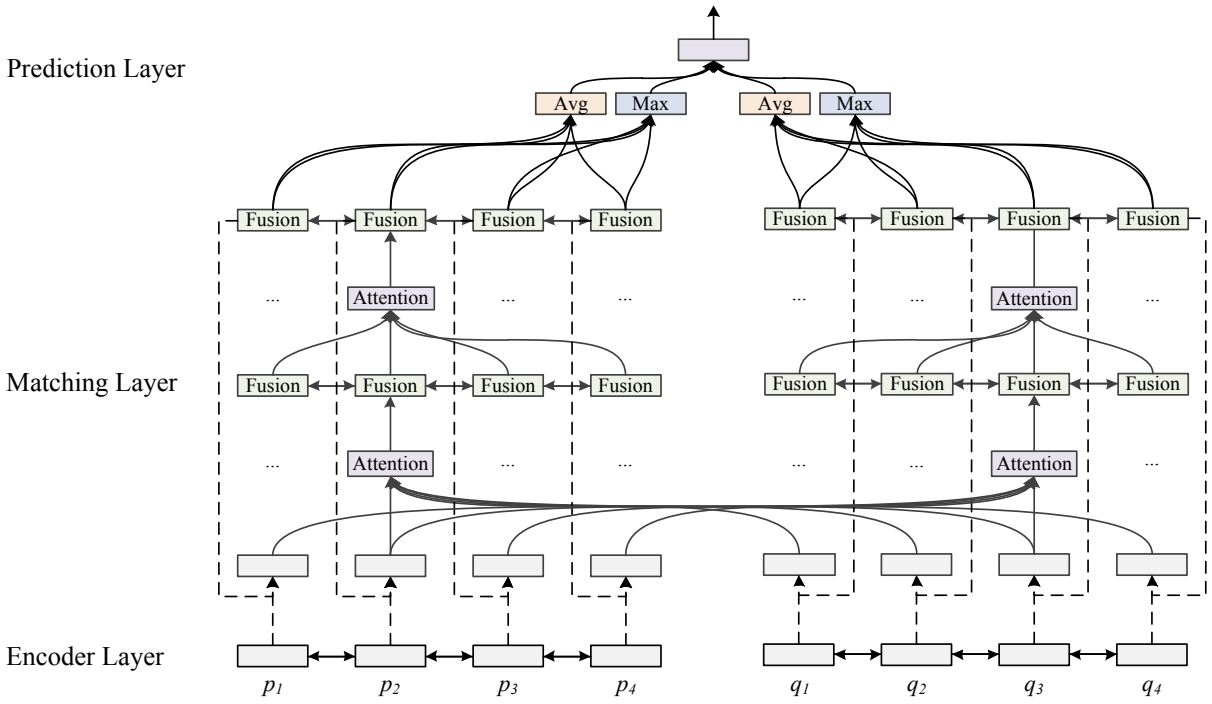

Figure 1: Architecture of AF-DMN. The dashed lines refer to the copy operation. In the first computational block of the matching layer, the input is from the encoder layer. After that, the input of computational blocks come from the previous computational blocks.

to learn the interaction of the sentence pair better.

- We evaluate our model on three challenging datasets and the results show that our model outperforms state-of-the-art baselines.

2 General Neural Model for Natural Language Inference

NLI is usually regarded as a classification task that predicts the relation $y \in Y$ for a given pair of sentences, where $Y = \{\text{entailment, contradiction, neutral}\}$. Recently, neural networks have been widely applied to this task because of their ability to assist with feature engineering [Bowman *et al.*, 2015; Wang and Jiang, 2015; Wang *et al.*, 2017b].

Formally, given two sentences $p = (p_1, ..., p_i, ..., p_m)$ and $q = (q_1, ..., q_j, ..., q_n)$, the aim is to predict the ground truth of relation y^*:

$$y^* = \arg\max_{y \in Y} P_r(y|p, q) \tag{1}$$

The $P_r(\cdot)$ is a neural network here. The general architecture of neural networks for NLI consists of three components: (1) **an encoder layer** converts the two sentences into semantic representations; (2) **a matching layer** aligns the information between the two sentences at the word level and produces new representations for the two sentences; and (3) **a prediction layer** predicts the relation of the given pair of sentences.

2.1 Encoder Layer

Given two sentences $p = (p_1, ..., p_i, ..., p_m)$ and $q = (q_1, ..., q_j, ..., q_n)$, the encoder layer first converts them into vectors $(\mathbf{e}_{p_1}, ..., \mathbf{e}_{p_i}, ..., \mathbf{e}_{p_m})$ and $(\mathbf{e}_{q_1}, ..., \mathbf{e}_{q_i}, ..., \mathbf{e}_{q_n})$ by looking up M respectively, where $M \in \mathbf{R}^{d \times |V|}$ is the embedding table. d is the dimension of embeddings and $|V|$ is the size of the vocabulary. The encoder layer then produces the semantic representation for each word in p and q using the Bi-directional Long Short-Term Memory (BiLSTM) neural network [Hochreiter and Schmidhuber, 1997]. Thus, each word in two sentences can be expressed as:

$$\mathbf{h}_{p_i} = \text{Bi-LSTM}(\mathbf{e}_{p_i}, \mathbf{h}_{p_{i-1}}, \mathbf{h}_{p_{i+1}}) \tag{2}$$

$$\mathbf{h}_{q_j} = \text{Bi-LSTM}(\mathbf{e}_{q_j}, \mathbf{h}_{q_{j-1}}, \mathbf{h}_{q_{j+1}}) \tag{3}$$

where, h_{p_0} and h_{q_0} are initialized as $\mathbf{0}$. Thus, the two sentences are converted to $\mathbf{H}_p = (\mathbf{h}_{p_1}, ..., \mathbf{h}_{p_i}, ..., \mathbf{h}_{p_m})$ and $\mathbf{H}_q = (\mathbf{h}_{q_1}, ..., \mathbf{h}_{q_j}, ..., \mathbf{h}_{q_n})$.

2.2 Matching Layer

Generally, the matching layer interacts with the information between two sentences for alignment. It can be formulated as:

$$\mathbf{V}_p = g(\mathbf{H}_p, \mathbf{H}_q), \ \mathbf{V}_q = g(\mathbf{H}_q, \mathbf{H}_p) \tag{4}$$

where $g(\cdot)$ is the matching function. Thus, $\mathbf{V}_p = (\mathbf{v}_{p_1}, ..., \mathbf{v}_{p_i}, ..., \mathbf{v}_{p_m})$ and $\mathbf{V}_q = (\mathbf{v}_{q_1}, ..., \mathbf{v}_{q_j}, ..., \mathbf{v}_{q_n})$ are the new representations for p and q respectively.

2.3 Prediction Layer

In this layer, a pooling layer is used to convert the vectors into a fixed-length vector and then feed it into a 2-layer multi-layer perception (MLP) classifier.

In order to capture all of the information and highlight the significant properties of the two sentences, we perform

a mean pooling and a max pooling on each of them and then concatenate them together:

$$\mathbf{V}_{p_{\text{mean}}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{v}_{p_i},\ \mathbf{V}_{p_{\text{max}}} = \max_{i=1}^{m}\mathbf{v}_{p_i} \tag{5}$$

$$\mathbf{V}_{q_{\text{mean}}} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{v}_{q_j},\ \mathbf{V}_{q_{\text{max}}} = \max_{j=1}^{n}\mathbf{v}_{q_j} \tag{6}$$

$$\mathbf{V} = [\mathbf{V}_{p_{\text{mean}}};\ \mathbf{V}_{p_{\text{max}}}\ ;\ \mathbf{V}_{q_{\text{mean}}};\ \mathbf{V}_{q_{\text{max}}}] \tag{7}$$

After obtaining the representation $\mathbf{V}$ of the two sentences, the distribution $P_r(\cdot)$ can be formalized as:

$$P_r(\cdot|p,q) = \text{softmax}(\mathbf{W}_2\tanh(\mathbf{W}_1 V + \mathbf{b}_1) + \mathbf{b}_2) \tag{8}$$

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

3 Attention-Fused Deep Matching Network for Natural Language Inference

Despite the success of conventional NLI models [Wang and Jiang, 2015; Chen *et al.*, 2017a; Wang *et al.*, 2017b], they only generate a shallow interaction between two sentences, which cannot sufficiently model the complex semantic interactions in the inference problem.

Inspired by the recent successful deep neural frameworks [He *et al.*, 2016; Wu *et al.*, 2016], we propose AF-DMN for natural language inference. AF-DMN exploits a more sophisticated matching layer, which consists of T computational blocks and each block has four sub-layers: (1) a cross attention layer; (2) a fusion layer for cross attention; (3) a self-attention layer; and (4) another fusion layer for self-attention. The AF-DMN will repeat the interaction process via stacked computational blocks T times.

3.1 Cross Attention

Cross attention captures the relevance between two sentences p and q. Concretely, in the t-th computational block, given the representations of two sentences computed in the previous block: $\mathbf{H}_p^{t-1} = (\mathbf{h}_{p_1}^{t-1}, ..., \mathbf{h}_{p_i}^{t-1}, ..., \mathbf{h}_{p_m}^{t-1})$ and $\mathbf{H}_q^{t-1} = (\mathbf{h}_{q_1}^{t-1}, ..., \mathbf{h}_{q_j}^{t-1}, ..., \mathbf{h}_{q_n}^{t-1})$, we first compute a co-attention matrix $\mathbf{A}^t \in \mathbf{R}^{m\times n}$. Each element $\mathbf{A}_{i,j}^t \in R$ indicates the relevance between the i-th word of sentence p and the j-th word of sentence q. Formally, the co-attention matrix could be computed as:

$$\mathbf{A}_{i,j}^t = {\mathbf{h}_{p_i}^{t-1}}^T \mathbf{W}^t \mathbf{h}_{q_j}^{t-1} + \langle \mathbf{U}_l^t, \mathbf{h}_{p_i}^{t-1}\rangle + \langle \mathbf{U}_r^t, \mathbf{h}_{q_j}^{t-1}\rangle \tag{9}$$

where $\mathbf{W}^t \in \mathbf{R}^{2h\times 2h}$, $\mathbf{U}_l^t, \mathbf{U}_r^t \in \mathbf{R}^{2h}$ are for the t-th computational block and $\langle\cdot,\cdot\rangle$ denotes the inter production operation. Then the attentive representation for each word p_i and q_j could be formalized as $\tilde{\mathbf{h}}_{p_i}^t \in \mathbf{R}^{2h}$ and $\tilde{\mathbf{h}}_{q_j}^t \in \mathbf{R}^{2h}$:

$$\mathbf{a}_{p_i}^t = \text{softmax}(\mathbf{A}_{i,:}^t),\ \mathbf{a}_{q_j}^t = \text{softmax}(\mathbf{A}_{:,j}^t) \tag{10}$$

$$\tilde{\mathbf{h}}_{p_i}^t = \mathbf{H}_q^{t-1}\cdot\mathbf{a}_{p_i}^t,\ \tilde{\mathbf{h}}_{q_j}^t = \mathbf{H}_p^{t-1}\cdot\mathbf{a}_{q_j}^t \tag{11}$$

3.2 Fusion for Cross Attention

In order to enhance the interaction further, we perform a fusion layer after the cross attention layer:

$$\bar{\mathbf{f}}_{p_i}^t = [\mathbf{h}_{p_i}^t; \tilde{\mathbf{h}}_{p_i}^t; \mathbf{h}_{p_i}^t - \tilde{\mathbf{h}}_{p_i}^t; \mathbf{h}_{p_i}^t \odot \tilde{\mathbf{h}}_{p_i}^t] \tag{12}$$

$$\tilde{\mathbf{f}}_{p_i}^t = \text{Relu}(\mathbf{W}_f^t \bar{\mathbf{f}}_{p_i}^t + \mathbf{b}_f^t) \tag{13}$$

$$\mathbf{f}_{p_i}^t = \text{Bi-LSTM}(\tilde{\mathbf{f}}_{p_i}^t, \mathbf{f}_{p_{i-1}}^t, \mathbf{f}_{p_{i+1}}^t) \tag{14}$$

where $[\cdot;\cdot;\cdot;\cdot]$ refers to the concatenation operation. Similarly, we derive the fusion result for sentence q as $\mathbf{f}_{q_j}^t$.

3.3 Self-Attention

In order to tackle the long-term dependency in a long sentence, we additionally introduce the self-attention mechanism after the cross attention layer.

Formally, for sentence p, we first compute a self-attention matrix $\mathbf{S}^t \in \mathbf{R}^{m\times m}$:

$$\mathbf{S}_{i,j}^t = \langle \mathbf{f}_{p_i}^t, \mathbf{f}_{p_j}^t\rangle \tag{15}$$

where $\mathbf{S}_{i,j}^t$ indicates the relevance between the i-th word and j-th word in sentence p.

Then the self attentive vector for each word can be computed as follow:

$$\mathbf{s}_{p_i}^t = \text{softmax}(\mathbf{S}_i^t),\ \bar{\mathbf{h}}_{p_i}^t = \mathbf{F}_p^t \cdot \mathbf{s}_{p_i}^t \tag{16}$$

where $\mathbf{F}_p^t = (\mathbf{f}_{p_1}^t, ..., \mathbf{f}_{p_i}^t, ..., \mathbf{f}_{p_m}^t)$, and $\mathbf{f}_{p_i}^t$ is computed as in Eq. (14).

We can similarly derive the self attentive vector for sentence q as $\bar{\mathbf{h}}_{q_j}^t$.

3.4 Fusion for Self-Attention

A fusion layer is introduced after the self-attention to enhance interaction. The fusion representation $\mathbf{h}_{p_i}^t$ of sentence p will be sent to the next block of the matching layer. Formally, $\mathbf{h}_{p_i}^t$ is computed as:

$$\bar{\mathbf{h}}_{p_i}^t = [\mathbf{f}_{p_i}^t; \bar{\mathbf{h}}_{p_i}^t; \mathbf{f}_{p_i}^t - \bar{\mathbf{h}}_{p_i}^t; \mathbf{f}_{p_i}^t \odot \bar{\mathbf{h}}_{p_i}^t] \tag{17}$$

$$\tilde{\mathbf{h}}_{p_i}^t = \text{Relu}(\mathbf{W}_h^t \bar{\mathbf{h}}_{p_i}^t + \mathbf{b}_h^t) \tag{18}$$

$$\mathbf{h}_{p_i}^t = \text{Bi-LSTM}(\tilde{\mathbf{h}}_{p_i}^t, \mathbf{h}_{p_{i-1}}^t, \mathbf{h}_{p_{i+1}}^t) \tag{19}$$

Similarly, we obtain the representation $\mathbf{h}_{q_j}^t$ for sentence q. Thus, in the t-th computational block, two sentences are converted to $\mathbf{H}_p^t = (\mathbf{h}_{p_1}^t, ..., \mathbf{h}_{p_i}^t, ..., \mathbf{h}_{p_m}^t)$ and $\mathbf{H}_q^t = (\mathbf{h}_{q_1}^t, ..., \mathbf{h}_{q_j}^t, ..., \mathbf{h}_{q_n}^t)$. Finally, $\mathbf{H}_p^t$ and $\mathbf{H}_q^t$ are sent to the prediction layer as input $\mathbf{V}_p$ and $\mathbf{V}_q$ after conducting the matching process T times.

4 Training

The object is to minimize the objective function $J(\Theta)$, which can be formulated as:

$$J(\Theta) = -\frac{1}{N}\sum_{i=1}^{N}\log P_r(y^{(i)}|p^{(i)}, q^{(i)}; \Theta) + \frac{1}{2}\lambda||\Theta||_2^2 \tag{20}$$

where N is the number of instances in the training set and $(p^{(i)}, q^{(i)})$ and $y^{(i)}$ are the sentence pair and the corresponding annotated label for the i-th instance respectively. Θ denotes all the trainable parameters of our model. We employ Adam [Kingma and Ba, 2014] as the optimizer.

	Train	Dev	Test	Avg.L		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
Quora	384K	10K	10K	12	12	107K

Table 1: Statistics of datasets: SNLI, MultiNLI, Quora. Avg.L refers to average length of two sentences. MultiNLI[1] and MultiNLI[2] indicate the in-domain and cross-domain versions respectively.

5 Experiments

5.1 Dataset

We evaluate our model on three datasets: the Stanford Natural Language Inference (SNLI), the MultiGenre NLI Corpus (MultiNLI) and Quora duplicate questions[1] (Quora). The detailed statistical information of datasets is shown in Table 1.

SNLI The SNLI corpus [Bowman *et al.*, 2015] contains 570,152 sentence pairs. Each pair is labeled with one of the following relationships: entailment, contradiction, or neutral. The data partition follows that of [Bowman *et al.*, 2015].

MultiNLI The MultiNLI 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. Since the MultiNLI corpus is collected from multiple domains, there are in-domain and cross-domain development/test sets.

Quora The Quora corpus contains over 400,000 question pairs. Each question pair is labeled with a binary value indicating whether the two questions are paraphrases of each other. In our experiment, we have the same partition as in [Wang *et al.*, 2017b].

5.2 Experiment Configuration

In our model, word embeddings and all hidden states of LSTMs and MLPs are 300 dimensions. For the SNLI dataset, there are 3 computational blocks in the deep matching layer, while there are 2 for MultiNLI and Quora datasets. We employ the Adam [Kingma and Ba, 2014] for training, whose default hyper-parameters β_1 and β_2 are set to 0.9 and 0.999 for optimization respectively. The initial learning rate of Adam is set to 0.0002. The learning rate is halved when the accuracy on the development set drops. We also employ a dropout strategy [Srivastava *et al.*, 2014] on word embeddings and all MLPs to avoid over-fitting. The dropout rate is set to 0.2. The batch size is set to 64. We set the maximum length of sentences to 200. For preprocessing, we just tokenize the sentences and lowercase the tokens.

For initialization, word embeddings are initialized with 300-dimensional GloVe vectors and updated during training. Other parameters include neural network parameters and Out Of Vocabulary (OOV) word embeddings are initialized randomly within [-0.01,0.01].

[1]https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

5.3 Ensemble

The ensemble strategy is an effective method to improve model accuracy. Following [Wang *et al.*, 2017b], our ensemble model averages the probability distributions from three individual single AF-DMNs, who have exactly identical architectures but distinguished initializations on parameters.

5.4 Overall Results

We use the accuracy to evaluate the performance of AF-DMN and other models on datasets SNLI, MultiNLI, and Quora.

SNLI Table 2 shows the results of different models on the training and test sets of SNLI. The baseline models in Table 2 can be categorized into two groups:

(1) *The first group* of models are based on sentence encoding. The TBCNN-pair neural model [Mou *et al.*, 2015] incorporates structural information into sentence representation. SPINN-PI [Bowman *et al.*, 2016] integrates tree-structured sentence interpretation into the linear sequential structure of a shift-reduce parser. Liu *et al.* [2016b] introduce the inner-attention that using a preliminary sentence representation to attend to words within the sentence. In [Munkhdalai and Yu, 2016], a memory augmented neural network is presented.

(2) *The second group* are based on attention-based models. Rocktäschel *et al.* [2015] extend the general sentence encoding model with attention while Wang and Jiang [2015] exploit long short-term memory (LSTM) for NLI. Liu *et al.* [2016a] pay more attention to the interaction of the text pair. Parikh *et al.* [2016] use attention to decompose the problem into subproblems that can be solved separately. Similar to [Wang and Jiang, 2015], Sha *et al.* [2016] design a new LSTM unit that takes the attention vector of one sentence as an inner state while reading the other sentence. Recently, Chen *et al.* [2017a] incorporate the chain LSTM and tree LSTM. Wang *et al.* [2017b] propose a bilateral multi-perspective matching.

In Table 2, the first three blocks are single models and the last two blocks are ensemble models. The proposed models, the single AF-DMN and the ensemble AF-DMN, achieve 88.6% and 89.0% on accuracy in SNLI test set respectively. Compared to previous work, AF-DMN outperforms previous models on both single and ensemble scenarios for natural language inference.

MultiNLI Table 3 shows the performance of different models on MultiNLI. Since this dataset aims to evaluate the quality of sentences representations, typical attention and memory models are not eligible for inclusion in this competition. As a result, most existing work on MultiNLI does not use the attention mechanism except for ESIM Chen *et al.* [2017b]. So far, ESIM is the strongest baseline which achieves a state-of-art performance on SNLI task. As [Chen *et al.*, 2017b] report, ESIM achieves 76.8% and 75.8% on the in-domain and cross-domain test sets of MultiNLI on accuracy respectively. The proposed model, AF-DMN, achieves 76.9% and 76.3% on the in-domain and cross-domain test sets on accuracy respectively. The results show that our model outperforms ESIM on both in-domain and cross-domain test sets.

Models	Train	Test
300D Tree-based CNN encoders [Mou *et al.*, 2015]	83.3	82.1
300D SPINN-PI encoders [Bowman *et al.*, 2016]M	89.2	83.2
600D (300+300) BiLSTM encoders with intra-attention [Liu *et al.*, 2016b]	84.5	84.2
300D NSE encoders [Munkhdalai and Yu, 2016]	86.2	84.6
100D LSTMs with attention [Rocktäschel *et al.*, 2015]	85.3	83.5
100D Deep fusion LSTM [Liu *et al.*, 2016a]	85.2	84.6
300D matching-LSTM [Wang and Jiang, 2015]	92.0	86.1
200D decomposable attention model with intra-sentence attention [Parikh *et al.*, 2016]	90.5	86.8
300D re-read LSTM [Sha *et al.*, 2016]	90.7	87.5
600D ESIM [Chen *et al.*, 2017a] (Single)	92.6	88.0
BiMPM [Wang *et al.*, 2017b] (Single)	90.9	87.5
AF-DMN (Single)	**94.5**	**88.6**
HIM (600D ESIM + 300D Syntactic tree-LSTM) [Chen *et al.*, 2017a] (Ensemble)	93.5	88.6
BiMPM [Wang *et al.*, 2017b] (Ensemble)	93.2	88.8
AF-DMN (Ensemble)	**94.9**	**89.0**

Table 2: Comparison with previous models on the SNLI dataset.

Models	In	Cross
ESIM [Chen *et al.*, 2017b]	76.8	75.8
AF-DMN	**76.9**	**76.3**

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

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	88.17
AF-DMN	**88.72**

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

Models	Dev	Test
Only cross attention	85.2	84.7
+ Fusion for cross attention	88.2	88.2
+ Self-attention	88.8	88.5
+ Fusion for self-attention (AF-DMN)	**89.1**	**88.6**

Table 5: Effect of components on the SNLI.

Num	Dev	Test
1	88.6	88.1
2	88.9	88.3
3	89.1	88.6

Table 6: Effect of number of blocks on the SNLI.

Quora Table 4 shows the performance of different models on the Quora test set. The baselines on Table 4 are all implemented in [Wang *et al.*, 2017b]. 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 an aggregation of convolution neural networks. As we can see, AF-DMN outperforms the baselines and achieves 88.72% in the test sets of the Quora corpus.

5.5 Effect of Components

To better understand the performance of AF-DMN, we analyze the effect of each key component of the proposed model. As illustrated in table 5, the first row is the AF-DMN without fusion layers and self-attention mechanism (only keeping cross attention layer) that we consider as the basic model. Compared to the full AF-DMN, the accuracy drops by 3.9% on test set of the SNLI dataset. By adding the fusion layer following the cross attention layer, the performance increases to 88.2%. That is a significant improvement and it is because the information from two sentences are gathered through the fusion layer. By additionally employing the self-attention mechanism, we achieve further improvement in accuracy. Finally, applying the fusion after the self-attention layer obtains the final accuracy. According to the results, all of the components positively contribute to the final performance.

Table 6 shows the performance with a different number of blocks. As we can see, with the number of blocks increases from 1 to 3, the performance increases both on the development set and the test set. Because of computational cost, we just set the number of blocks as 3 on SNLI.

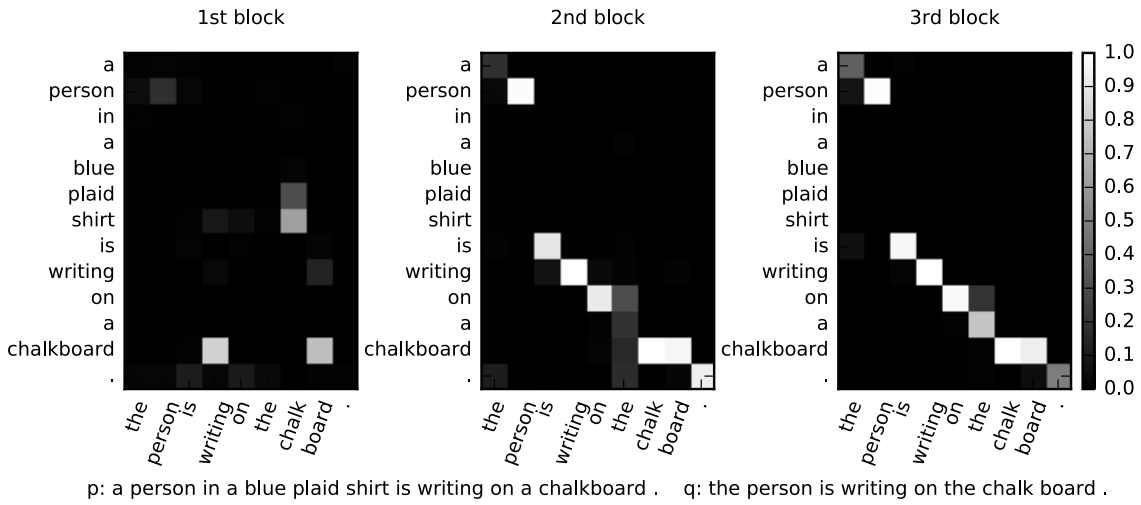

Figure 2: The three sub-figures from left to right are visualizations of the cross attention matrices in the 1st, 2nd and 3rd block respectively.

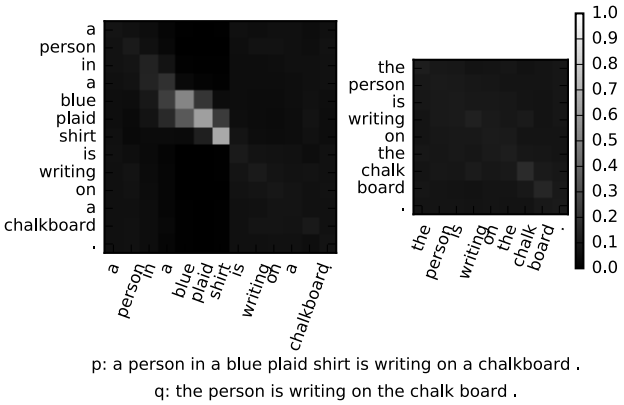

Figure 3: Visualization of the self-attention matrices of two sentences in the 3rd block.

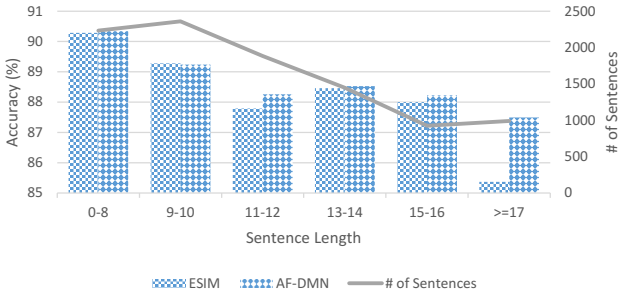

Figure 4: Performance on sentence pairs with different sentence lengths on the test set of SNLI. The histogram is evaluated by the left y-axis and the line is evaluated by the right y-axis.

5.6 Case Study

As depicted in Figure 2, this is an instance from the test set of the SNLI dataset: {p: a person in a blue plaid shirt is writing on a chalk board. q: the person is writing on the chalk board. The label y: entailment.}. The results are produced by AF-DMN with 3 computational blocks in the deep matching layer, demonstrating the changes of cross attention from low block to high block in the matching layer. The three sub-figures from left to right show the alignment results in the 1st, 2nd, 3rd computational block of the matching layer respectively. With the increment of interaction, the alignment of the sentence pair becomes more clear and accurate. The highlighted cells imply that cross attention aims to figure out the alignments between the two sentences.

Figure 3 is the visualization of the self-attention layer in the last block for the same test instance. The left sub-figure refers to sentence p and the right sub figure refers to sentence q. In the left sub figure, the phrase "in a blue plaid shirt" is highlighted, which is the inconsistent part between the two sentences and also the critical factor that determines if p can entail q. It shows that the proposed AF-DMN is capable of capturing key information between a pair of sentences for natural language inference.

5.7 Effect of Sentence Length

Since ESIM is currently the state-of-the-art model for the SNLI dataset and the code is available[2], we would like to compare our model with ESIM on the SLNI dataset in this sub sections.

Sentence length is one of the most important factors that can affect the performance of neural models. To further analyze the performance of the proposed model, we evaluate the performance of AF-DMN and ESIM on sentence pairs of different text lengths on the SNLI test set. Specifically, we use the floor average length of a pair of sentences (p and q) as the length of the sentence pair. As illustrated in Figure 4, for sentence pairs with a length less than or equal to 8, both of the models achieve the best performance that is better than 90%. However, with the incremental increases in length of the sentence pairs, the performance of both models decreases. Throughout the results, except for sentence pairs with a length range from 9 to 10, AF-DMN is defeated with

[2]https://github.com/lukecq1231/nli

a small gap, the proposed model outperforms ESIM on rest length. When the length of sentence pair is greater than or equal to 17, our model outperforms ESIM by a large margin on this length. It indicates that the proposed model benefits from the self-attention mechanism that has an advantage in incorporating local context information within a sentence. Meanwhile, the deep matching process is able to extract key information from sentences multiple times, thereby benefiting cases on long sentences.

6 Related Works

Natural language inference (NLI) has been widely investigated for many years. Previous work on NLI relies on handcrafted features such as n-gram overlapping, syntactic information and so on. Heilman and Smith [2010] propose an effective tree edit approach to model relations between sentence pairs. Bowman *et al.* [2015] adopt a lexicalized classifier which implements features (BLEU score, n-gram overlap, etc.) for SNLI. All above mentioned methods perform reasonably well for a specific task but are difficult to generalize for others.

Benefiting from the development of deep learning and the availability of large-scale annotated datasets [Bowman *et al.*, 2015; Williams *et al.*, 2017], data-driven models attract more attentions. Liu *et al.* [2016b] introduce inner-attention using a preliminary sentence representation to attend words within the sentence. In [Munkhdalai and Yu, 2016], a memory augmented neural network for NLI is presented. All of these models are based on sentence encoding. However, they neglect the interaction between sentences. To address the problem, Wang and Jiang [2015] design a special LSTM called matching-LSTM which performs word-by-word matching of the hypothesis with the premise. Furthermore, Cui *et al.* [2016]; Wang *et al.* [2017b]; Chen *et al.* [2017a] propose a new framework to model the relationship between two sentences, which performs the matching on pairs of sentences in two directions. Besides cross attention, because of the limitations of the RNN model on the long-term dependency problem, the self-attention mechanism is proposed, which aims to align the sequence with itself and has been used in variety of tasks [Lin *et al.*, 2017; Wang *et al.*, 2017a]. The self-attention mechanism can capture contextual information from the whole sentence.

For the proposed AF-DMN, it performs cross attention and self-attention on a sentence pair for multiple iterations through the stacked computational blocks. Based on this mechanism, we obtain representations of two sentences with multiple levels of abstraction and achieve better performance on several challenging datasets.

7 Conclusions and Future Work

In this paper, we propose an attention-fused deep matching network (AF-DMN) for natural language inference. It leverages cross attention and self-attention jointly. We evaluate our model on three datasets: SNLI, MultiNLI, and Quora duplicate questions. Experiment results show that AF-DMN achieves state-of-the-art performance. In the future, we will further investigate whether unlabeled data can help to learn more accurate sentence representations and relationships between inputs, to ameliorate data sparseness.

Acknowledgments

We are grateful to the anonymous reviewers. The work of this paper is funded by the (key) project of National Natural Science Foundation of China (No. 91520204, 61572154) and the project of National High Technology Research and Development Program of China (863 Program) (No. 2015AA015405). This project is also partially funded by Microsoft Research Asia.

References

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

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

[Chen *et al.*, 2017a] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *Proc. ACL*, 2017.

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

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Cui *et al.*, 2016] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423*, 2016.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Heilman and Smith, 2010] Michael Heilman and Noah A Smith. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019. Association for Computational Linguistics, 2010.

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

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

[Lin *et al.*, 2017] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[Liu *et al.*, 2016a] Pengfei Liu, Xipeng Qiu, Jifan Chen, and Xuanjing Huang. Deep fusion lstms for text semantic matching. In *ACL (1)*, 2016.

[Liu *et al.*, 2016b] Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*, 2016.

[MacCartney, 2009] Bill MacCartney. *Natural language inference*. Stanford University, 2009.

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

[Munkhdalai and Yu, 2016] Tsendsuren Munkhdalai and Hong Yu. Neural tree indexers for text understanding. *arXiv preprint arXiv:1607.04492*, 2016.

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

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

[Sha *et al.*, 2016] Lei Sha, Baobao Chang, Zhifang Sui, and Sujian Li. Reading and thinking: Re-read lstm unit for textual entailment recognition. In *COLING*, pages 2870–2879, 2016.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[Tan *et al.*, 2015] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*, 2015.

[Wang and Jiang, 2015] Shuohang Wang and Jing Jiang. Learning natural language inference with lstm. *arXiv preprint arXiv:1512.08849*, 2015.

[Wang *et al.*, 2017a] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 189–198, 2017.

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

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

[Wu *et al.*, 2016] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.