Multiway Attention Networks for Modeling Sentence Pairs

Chuanqi Tan[†*], Furu Wei[‡], Wenhui Wang[+], Weifeng Lv[†], Ming Zhou[‡]
[†]State Key Laboratory of Software Development Environment, Beihang University, China
[‡]Microsoft Research, Beijing, China
[+]Peking University, Beijing, China
tanchuanqi@nlsde.buaa.edu.cn, fuwei@microsoft.com, wangwenhui@pku.edu.cn
lwf@buaa.edu.cn, mingzhou@microsoft.com

Abstract

Modeling sentence pairs plays the vital role for judging the relationship between two sentences, such as paraphrase identification, natural language inference, and answer sentence selection. Previous work achieves very promising results using neural networks with attention mechanism. In this paper, we propose the multiway attention networks which employ multiple attention functions to match sentence pairs under the matching-aggregation framework. Specifically, we design four attention functions to match words in corresponding sentences. Then, we aggregate the matching information from each function, and combine the information from all functions to obtain the final representation. Experimental results demonstrate that the proposed multiway attention networks improve the result on the Quora Question Pairs, SNLI, MultiNLI, and answer sentence selection task on the SQuAD dataset.

Paraphrase Identification
S: she struck a deal with RH to pen a book today
$+$: she signed a contract with RH to write a book
$-$: she denied today that she struck a deal with RH
Natural Language Inference
S: children smiling and waving at camera
E: there are children present
C: the kids are frowning
N: they are smiling at their parents
Answer Sentence Selection
Q: how are glacier caves formed ?
$+$: a glacier cave is a cave formed within the ice of a glacier.
$-$: the ice facade is approximately 60m high

Table 1: For paraphrase identification, $+$ means it is the paraphrase of S, otherwise $-$. For natural language inference, E, C, and N mean entailment, contradiction, and neutral, respectively. For answer sentence selection, $+$ means it can answer the question Q, otherwise $-$.

1 Introduction

In this paper, we study the task of modeling a pair of sentences, which aims to compare two sentences and identify the relationship between them. It is a fundamental technology for a variety of tasks. For example, in the paraphrase identification task, it is used to determine whether two sentences are paraphrase or not [Madnani *et al.*, 2012]. In the natural language inference task, it is utilized to judge whether a hypothesis sentence can be inferred from a premise sentence [Bowman *et al.*, 2015]. In the answer sentence selection task, it is employed to assess the relevance between question-answer pairs and rank all candidate answer sentences [Yang *et al.*, 2015]. Table 1 shows corresponding examples of abovementioned three tasks.

Previous works using neural networks with attention mechanism show effectiveness on this task. These methods can be organized into two frameworks. The first framework is to model sentence pairs by encoding each sentence separately and then making the decision based on the two representations [Yu *et al.*, 2014; Bowman *et al.*, 2016]. The limitation of this framework is that two sentences do not interact during the encoding part. Some methods apply the attention mechanism to improve the interaction of two sentences [Tan *et al.*, 2016]. However, it usually uses the representation of one sentence to attend another sentence, which still works at sentence level but lacks word level interactions. The second framework is based on the matching-aggregation framework [Wang and Jiang, 2016a; Wang *et al.*, 2017b]. It applies the attention mechanism at the word level to improve the matching between words in two sentences. Then the matching information is aggregated to the sentence level for making the decision. This framework enables the word level interaction and leads to promising results. Inspired by this framework, we consider that the word level matching is very important for modeling sentence pairs.

To this end, we propose a multiway attention network (MwAN) for modeling sentence pairs. We propose using multiple attention functions to match two sentences at the word level. Specifically, we use four kinds of attention functions, including the concatenated attention function which is used in Rocktäschel *et al.* [2015] for natural language infer-

[*] Contribution during internship at Microsoft Research Asia.

ence, and the bilinear attention function which is utilized in Chen *et al.* [2016] to match the question and passage in the reading comprehension. In addition to these two widely-used attention functions, we use two other attention functions that calculate the word relation by the element-wise dot product and difference of two vectors. Such element-wise comparisons are conducted on distributed representations of different text granularities in previous works [Wang and Jiang, 2016a; Chen *et al.*, 2017] to model the interactions between words and sentences. We directly apply these two functions between words, which enhances the word level relation modeling.

Our model is built on the matching-aggregation framework. Given two sentences, we encode them with a bi-directional RNN to obtain contextual word representation of words in two sentences based on the word embeddings. Then we apply four above-mentioned attention functions to match two sentences at the word level. Next, we aggregate the matching information from multiway attention functions with two steps. The matching information of each function is first aggregated by a bi-directional RNN along with all words. Then the outputs of all attention functions are adaptively combined by applying the attention mechanism. We apply another bi-directional RNN to pass through the combined representation to aggregate the mixture matching information. Finally, we apply the attention-pooling to the matching information for a fix-length vector and feed it into a multilayer perceptron for the final decision.

We conduct experiments on three tasks with four standard benchmark datasets. Experimental results show that our multiway attention network achieves state-of-the-art results on the Quora Question Pairs dataset for paraphrase identification, the SNLI and MultiNLI datasets for natural language inference, and the SQuAD dataset for answer sentence selection. We also conduct ablation tests by using single attention function and removing any of functions. Results justify that all of four attention functions help the model performance. Moreover, we observe that different attention functions are good at samples of different categories, which explains why our framework improves the result since it combines the strengths from multiple attention functions.

2 Related Work

Recently, deep learning and neural network models achieve promising results in modeling sentence pairs. Basically, two sentences are encoded into sentence vectors with a neural network encoder, and then the relationship between two sentences was decided solely based on the two sentence vectors [Bowman *et al.*, 2015; Yang *et al.*, 2015; Tan *et al.*, 2016]. However, this kind of framework ignores the lower level interaction between two sentences.

The matching-aggregation framework is therefore proposed matching two sentences at the word level and then aggregating the matching information based on the attention mechanism for the final decision. Rocktäschel *et al.* [2015] use the attention-based technique to improve the performance of LSTM-based recurrent neural network. They employ the word-by-word attention to obtain a sentence-pair encoding from fine-grained reasoning via soft-alignment of words and phrases in the premise and hypothesis, which achieves very promising result on the SNLI dataset. Wang and Jiang [2016b] propose match-LSTM for the natural language inference that tries to match the current word in the hypothesis with an attention-weighted representation of the premise calculated by the word-by-word attention. Wang *et al.* [2017b] propose BiMPM that matches two sentences by a bilateral matching with attention mechanism in multiple perspectives, which achieves the state-of-the-art results on the paraphrase identification, natural language inference, and answer sentence selection. They propose four types of representation instead of the attention-weighted representation to improve the result. Moreover, Cheng *et al.* [2016] propose LSTMN, which improves the attention mechanism by a memory network for the natural language inference. Parikh *et al.* [2016] match two sentences by attending, comparing, and aggregating for the SNLI dataset. Sha *et al.* [2016] propose re-reading two sentences with the attention mechanism to improve the memory of the other sentence for a better understanding. Munkhdalai and Yu [2016] propose using the tree structure to improve the recurrent or recursive architecture for the natural language inference and answer sentence selection. Chen *et al.* [2017] incorporate syntactic parsing information to basic attention-based neural networks, which achieves the state-of-the-art result on the SNLI dataset.

Furthermore, CNN is also applied on the sentence pairs modeling and achieves very promising results [Gong *et al.*, 2017]. Yin *et al.* [2016] propose the attention-based CNN for the paraphrase identification, natural language inference, and answer sentence selection. Wang and Jiang [2016a] use the CNN to aggregate the matching information for the answer sentence selection.

Our model belongs to the matching-aggregation framework. Based on the attention mechanism, we propose using multiple functions to match words in two sentences, and make the decision by aggregating the matching information from multiple attention functions.

3 Task Definition

Formally, we can represent each example of sentence pairs as a triple (Q, P, y), where $Q = (q_1, ..., q_i, ..., q_N)$ is a sentence with a length N, $P = (p_1, ..., p_j, ..., p_M)$ is another sentence with a length M, and $y \in \mathcal{Y}$ is the label representing the relationship between Q and P.

Specifically, for a paraphrase identification task, Q and P are two sentences, $\mathcal{Y} = \{0, 1\}$, where $y = 1$ means that Q and P are paraphrase of each other, and $y = 0$ otherwise.

For a natural language inference task, Q is a premise sentence, P is a hypothesis sentence, and $\mathcal{Y}$ =$\{$entailment, contradiction, neutral$\}$ where entailment indicates P can be inferred from Q, contradiction indicates P cannot be true condition on Q, and neutral means P and Q are irrelevant to each other.

For an answer sentence selection task, Q is a question, P is a candidate answer sentence, and $\mathcal{Y} = \{0, 1\}$ where $y = 1$ means P is a correct answer sentence for Q, and $y = 0$ otherwise.

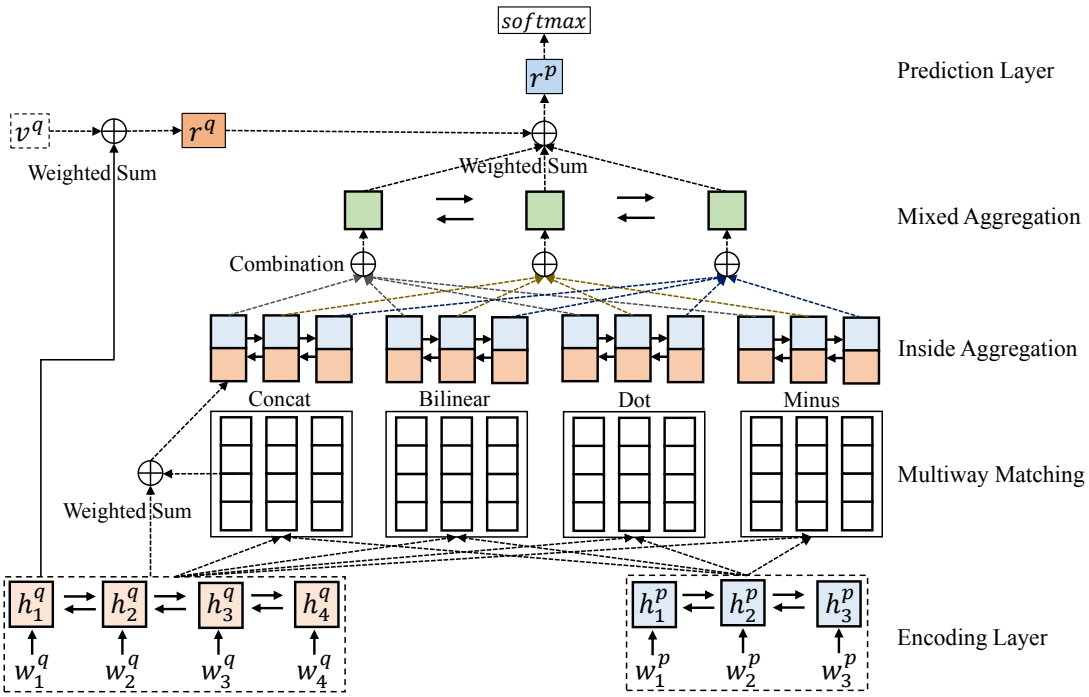

Figure 1: Overview of Multiway Attention Networks.

4 Our Approach

We show the multiway attention networks in Figure 1. It consists of five parts under the matching-aggregation framework. Specifically, we obtain contextual word representation for two sentences using a bi-directional RNN. Then we match words between two sentences in multiple ways. For every word pair from Q and P, we can obtain four matching scores using multiple attention functions. Next, we aggregate the matching information along with words in P in two steps. We first match two sentences inside each attention function, and then combine the matching information from all functions. The bi-directional RNN is applied to aggregate the matching information both in the inside aggregation and mixed aggregation. Finally, we use the attention pooling to aggregate the matching information in P for a fix-length vector and apply a multilayer perceptron (MLP) classifier for the final decision.

4.1 Gated Recurrent Unit

We use Gated Recurrent Unit (GRU) [Cho *et al.*, 2014] instead of basic RNN. Equation 1 describes the mathematical model of the GRU. r_t and z_t are the gates and h_t is the hidden state.

$$z_t = \sigma(W_{hz}h_{t-1} + W_{xz}x_t + b_z) \tag{1a}$$
$$r_t = \sigma(W_{hr}h_{t-1} + W_{xr}x_t + b_r) \tag{1b}$$
$$\hat{h_t} = \Phi(W_h(r_t \odot h_{t-1}) + W_x x_t + b) \tag{1c}$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h_t} \tag{1d}$$

4.2 Encoding Layer

Consider two sentences Q = $\{w_t^q\}_{t=1}^N$ and P = $\{w_t^p\}_{t=1}^M$, we first convert the words to their respective word-level embeddings and contextual embeddings. The contextual embeddings are generated by taking the output of a pre-trained language model [Peters *et al.*, 2018], which has shown the effectiveness on many NLP tasks. We then use a bi-directional GRU to produce new representation $h_1^q, \ldots, h_N^q$ and $n_1^p, \ldots, h_M^P$ of all words in two sentences respectively:

$$w_t^q = [e_t^q, lm_t^q] \tag{2a}$$
$$\overrightarrow{h}_t^q = \text{GRU}(\overrightarrow{h}_{t-1}^q, w_t^q) \tag{2b}$$
$$\overleftarrow{h}_t^q = \text{GRU}(\overleftarrow{h}_{t+1}^q, w_t^q) \tag{2c}$$

and $h_t^q = [\overrightarrow{h}_t^q, \overleftarrow{h}_t^q]$. Meanwhile, we use another bi-directional GRU to encode each w_t^p to the representation h_t^p.

4.3 Multiway Matching

Previous works show the effectiveness of the word-level attention in modeling sentence pairs [Rocktäschel *et al.*, 2015; Wang and Jiang, 2016b]. In our model, we design multiple attention functions to compare two vectors of the words in two sentences, namely concat attention, bilinear attention, dot attention, and minus attention. Therefore, at each position t of P, the word can match Q using four attention functions to obtain corresponding weighted-sum representations of Q.

$$q_t^k = f_k(h^q, h_t^p, W_k) \tag{3}$$

where h_t^p is the representation of P at the position t, h^q is the representation of all positions in Q, and q_t^k is the corresponding representation of Q using the attention function f_k based on the parameter W_k, where $k = (c, b, d, m)$, which represents the concat attention, bilinear attention, dot attention, and minus attention, respectively.

Concat Attention :

$$s_j^t = {v_c}^T tanh(W_c^1 h_j^q + W_c^2 h_t^p) \quad (4a)$$
$$a_i^t = exp(s_i^t)/\Sigma_{j=1}^N exp(s_j^t) \quad (4b)$$
$$q_t^c = \Sigma_{i=1}^N a_i^t h_i^q \quad (4c)$$

Bilinear Attention :

$$s_j^t = h_j^{qT} W_b h_t^p \quad (5a)$$
$$a_i^t = exp(s_i^t)/\Sigma_{j=1}^N exp(s_j^t) \quad (5b)$$
$$q_t^b = \Sigma_{i=1}^N a_i^t h_i^q \quad (5c)$$

Dot Attention :

$$s_j^t = {v_d}^T tanh(W_d(h_j^q \odot h_t^p)) \quad (6a)$$
$$a_i^t = exp(s_i^t)/\Sigma_{j=1}^N exp(s_j^t) \quad (6b)$$
$$q_t^d = \Sigma_{i=1}^N a_i^t h_i^q \quad (6c)$$

Minus Attention :

$$s_j^t = {v_m}^T tanh(W_m(h_j^q - h_t^p)) \quad (7a)$$
$$a_i^t = exp(s_i^t)/\Sigma_{j=1}^N exp(s_j^t) \quad (7b)$$
$$q_t^m = \Sigma_{i=1}^N a_i^t h_i^q \quad (7c)$$

4.4 Aggregation

We aggregate the matching information from multiway attention functions in two steps. The inside aggregation is to aggregate the matching information along with words in the sentence P inside each attention function. For each position t, we concatenate the word representation h_t^p in P with its corresponding representation q_t^k of Q [Wang and Jiang, 2016b] and add a gate to determine the importance of the concatenated representation [Wang *et al.*, 2017a]. Then we use the bi-directional GRU to pass through each position in P. We take the concat attention as an example,

$$x_t^c = [q_t^c, h_t^p] \quad (8a)$$
$$g_t = sigmoid(W_g x_t^c) \quad (8b)$$
$$x_t^{c*} = g_t \odot x_t^c \quad (8c)$$
$$\overrightarrow{h}_t^c = \text{GRU}(\overrightarrow{h}_{t-1}^c, x_t^{c*}) \quad (8d)$$
$$\overleftarrow{h}_t^c = \text{GRU}(\overleftarrow{h}_{t+1}^c, x_t^{c*}) \quad (8e)$$

and $h_t^c = [\overrightarrow{h}_t^c, \overleftarrow{h}_t^c]$. For the bilinear, dot, and minus attention, we obtain the h_t^b, h_t^d, and h_t^m, respectively.

The mixed aggregation is to combine the matching information from all attention functions. We apply the attention mechanism to adaptively combine four representations with the parameter v^a as the input.

$$s_j = v^T tanh(W^1 h_t^j + W^2 v^a)(j = c, b, d, m) \quad (9a)$$
$$a_i = exp(s_i)/\Sigma_{j=(c,b,d,m)} exp(s_j) \quad (9b)$$
$$x_t = \Sigma_{i=(c,b,d,m)} a_i h_t^i \quad (9c)$$

Then, we feed x_t into a bi-directional GRU to aggregate the matching information from multiple attention functions. We obtain $h_t^o = [\overrightarrow{h}_t^o, \overleftarrow{h}_t^o]$ for all positions in P.

$$\overrightarrow{h}_t^o = \text{GRU}(\overrightarrow{h}_{t-1}^o, x_t) \quad (10a)$$
$$\overleftarrow{h}_t^o = \text{GRU}(\overleftarrow{h}_{t+1}^o, x_t) \quad (10b)$$

4.5 Prediction Layer

After aggregating the information from multiway matching, we convert the resulting representations of all positions in P to a fixed-length vector with pooling and feed it into a classifier to determine the relation between two sentences. We first apply an attention pooling with the parameter v^q to select the important information in Q.

$$s_j = v^T tanh(W_q^1 h_j^q + W_q^2 v^q) \quad (11a)$$
$$a_i = exp(s_i)/\Sigma_{j=1}^N exp(s_j) \quad (11b)$$
$$r^q = \Sigma_{i=1}^N a_i h_i^q \quad (11c)$$

Then we use this representation r^q to select the information in the matching vectors.

$$s_j = v^T tanh(W_p^1 h_j^o + W_p^2 r^q) \quad (12a)$$
$$a_i = exp(s_i)/\Sigma_{j=1}^M exp(s_j) \quad (12b)$$
$$r^p = \Sigma_{i=1}^M a_i h_i^o \quad (12c)$$

Finally, we feed r^p into a multilayer perceptron (MLP) classifier for the probability p_i of each label in the corresponding task.

For all tasks, the objective function is to minimize the following cross entropy:

$$\mathcal{L} = -\sum_{i=1}^{N} [y_i \log p_i + (1 - y_i) \log(1 - p_i)] \quad (13)$$

where y_i denotes a label, in paraphrase detection it is $(0, 1)$ to represent whether two sentences are paraphrase, in natural language inference it is the relation of two sentences of entailment, contradiction, and neutral, in answer sentence selection it is $(0, 1)$ to represent whether P can answer the question Q.

4.6 Implementation Details

We use 300-dimensional uncased pre-trained *GloVe* embeddings [Pennington *et al.*, 2014] without update during training. We use zero vectors to represent all out-of-vocabulary words. Hidden vector length is set to 150 for all layers. We apply dropout [Srivastava *et al.*, 2014] between layers, with dropout rate 0.2. The model is optimized using AdaDelta [Zeiler, 2012] with initial learning rate of 1.0.

5 Experiment

We conduct experiments on three tasks with four datasets. Experimental results show that our model outperforms our baseline and other competing approaches. We also conduct ablation tests to analyze the contribution of each attention function.

5.1 Dataset and Evaluation Metrics

Quora Question Pairs This dataset consists of over 400,000 question pairs, and each question pair is annotated with a binary value indicating whether the two questions are paraphrase of each other. We select 5,000 paraphrases and 5,000 non-paraphrases as the development set, and use another 5,000 paraphrases and 5,000 non-paraphrases as the test set. We keep the remaining instances as the training set.

Method	Dev	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	88.69	88.17
DIIN	89.44	89.06
MwAN	**89.60**	**89.12**

Table 2: Results for paraphrase identification on the Quora dataset.

SNLI It is a natural language inference dataset [Bowman *et al.*, 2015]. The original data set contains 570,152 sentence pairs, each labeled with one of the following relationships: entailment, contradiction, neutral and −, where − indicates a lack of human annotation and is usually discarded [Wang and Jiang, 2016b]. In the end, we have 549,367 pairs for training, 9,842 pairs for development and 9,824 pairs for test.

MultiNLI It is a natural language inference dataset [Williams *et al.*, 2017]. Models can be evaluated on both the matched test examples, which are derived from the same sources as those in the training set, and the mismatched examples, which do not closely resemble any seen at training time. This dataset contains 392,702 pairs for training, 9,815 matched pairs and 9,832 mismatched pairs for development, 9,796 matched pairs and 9,847 mismatched pairs for test.

SQuAD It is a reading comprehension dataset, where the answer to each question is a span of text from the corresponding passage [Rajpurkar *et al.*, 2016]. To evaluate our answer selection task, we split the sentences from the passage using the Stanford CoreNLP Toolkit [Manning *et al.*, 2014] and treat the sentence which contains the correct span as the answer sentence. As the author of SQuAD only publishes the training set and the development set, we split the 10,570 instances in the development set to 5,000 for development and 5,570 for test. We use mean reciprocal rank (MRR) for evaluation.

5.2 Result on Paraphrase Identification

We compare our model with several baselines shown in Table 2. Siamese-CNN and Siamese-LSTM [Wang *et al.*, 2016a] are based on the sentence encoding framework. Multi-Perspective-CNN and Multi-Perspective-LSTM improve them by multiple perspective cosine matching functions [Wang *et al.*, 2017b]. We also compare our MwAN with models under the matching-aggregation framework, such as LDC [Wang *et al.*, 2016b] and BiMPM [Wang *et al.*, 2017b] under RNN and DIIN [Gong *et al.*, 2017] under CNN. Our MwAN outperforms all baselines with 89.60% accuracy on the development set and 89.12% accuracy on the test set.

5.3 Result on Natural Language Inference

Table 3 shows results on the SNLI. Our MwAN achieves the state-of-the-art result with 88.3% test accuracy. Moreover, we also report the ensemble result. Following the ensemble strategy in Wang *et al.* [2017b], we train the model 4 times with the same setting, and sum the probability of each single model to decide the result. The test accuracy is 89.4%, which outperforms all baselines and achieves the best test score on the SNLI benchmark to date.

Method	Train	Test
LSTM with attention [Rocktäschel *et al.*, 2015]	85.3	83.5
mLSTM [Wang and Jiang, 2016b]	92.0	86.1
LSTMN [Cheng *et al.*, 2016]	88.5	86.3
decomposable attention [Parikh *et al.*, 2016]	89.5	86.3
Intra-sentence attention [Parikh *et al.*, 2016]	90.5	86.8
NTI-SLSTM-LSTM [Munkhdalai and Yu, 2016]	88.5	87.3
re-read LSTM [Sha *et al.*, 2016]	90.7	87.5
BiMPM [Wang *et al.*, 2017b]	90.9	87.5
btree-LSTM encoders [Parikh *et al.*, 2016]	88.6	87.6
DIIN [Gong *et al.*, 2017]	91.2	88.0
ESIM [Chen *et al.*, 2017]	92.6	88.0
ESIM+Syntactic tree-LSTM [Chen *et al.*, 2017]	93.5	88.6
ESIM+Elmo [Peters *et al.*, 2018]	-	88.7
BiMPM ensemble [Wang *et al.*, 2017b]	93.2	88.8
DIIN ensemble [Gong *et al.*, 2017]	92.3	88.9
MwAN	94.5	88.3
MwAN (ensemble)	**95.5**	**89.4**

Table 3: Results for natural language inference on the SNLI dataset.

Method	Matched	Mismatched
CBOW	64.8	64.5
BiLSTM	66.9	66.9
ESIM	72.3	72.1
DIIN	78.8	77.8
DIIN (ensemble)	**80.0**	78.7
TALP-UPC*	67.9	68.2
LCT-MALTA*	70.7	70.8
Rivercorners*	72.1	72.1
Rivercorners (ensemble)*	72.2	72.8
alpha*	73.5	73.6
YixinNie-UNC-NLP*	74.5	73.5
alpha (ensemble)*	74.9	74.9
MwAN	78.5	77.7
MwAN (ensemble)	79.8	**79.4**

Table 4: Results for natural language inference on the MultiNLI test set. *indicates that the model participating in the competition on June 15th, 2017.

Table 4 shows results on the MultiNLI. Results of baselines are taken from the official report [Williams *et al.*, 2017]. Our MwAN achieves the state-of-the-art results of 78.5% and 77.7% on the Matched and Mismatched pairs, respectively. An ensemble model achieves the state-of-the-art result with 79.8% accuracy on the Matched pairs, and outperforms all baselines with 79.4% accuracy on the Mismatched pairs.

5.4 Result on Answer Sentence Selection

For the answer sentence selection task, we do not use the WikiQA dataset because it is too small to train a complex deep learning model. We implement two basic deep learning baselines Bigram-CNN [Yu *et al.*, 2014] and Attentive LSTM [Tan *et al.*, 2016], and we also report results of two state-of-the-art methods BiMPM [Wang *et al.*, 2017b] and CNN-MULT [Wang and Jiang, 2016a]. All these baselines achieve very promising results on the WikiQA dataset for this task. As shown in Table 6, our MwAN achieves 91.35 in terms of MRR, which outperforms all baseline methods.

Method	Quora		SNLI		MultiNLI		SQuAD	
	Dev	Test	Dev	Test	Matched	Mismatched	Dev	Test
MwAN	**89.60**	**89.12**	**88.58**	**88.30**	**78.78**	**78.45**	**91.45**	**91.35**
Concat attention	87.10	86.54	88.02	87.75	76.96	76.83	90.44	90.17
Bilinear attention	87.19	86.57	87.99	87.69	77.57	77.51	90.39	90.10
Dot attention	89.04	88.34	88.05	87.74	78.04	77.83	90.49	90.30
Minus attention	86.99	86.65	88.01	87.62	77.04	76.53	90.38	90.02
w/o Concat attention	89.06	88.62	88.36	88.15	78.36	78.01	90.80	90.59
w/o Bilinear attention	89.38	88.43	88.20	88.05	78.37	78.27	90.94	90.54
w/o Dot attention	87.50	86.96	88.40	88.11	77.83	77.71	90.62	90.45
w/o Minus attention	89.36	88.49	88.36	88.08	78.34	78.21	91.06	90.82

Table 5: Ablation experiments on four datasets.

Method	MRR
Bigram-CNN	79.18
Attentive LSTM	82.24
BiMPM	90.37
CNN-MULT	90.72
MwAN	**91.35**

Table 6: Results for answer sentence selection on the SQuAD dataset.

Dataset	Concat	Bilinear	Dot	Minus
Quora (All)	0.14	0.20	0.52	0.14
Quora (+)	0.12	0.17	**0.59**	0.12
Quora (-)	**0.16**	**0.23**	0.44	**0.17**
SNLI (All)	0.11	0.14	0.66	0.09
SNLI (E)	0.06	0.11	**0.77**	0.06
SNLI (C)	0.09	**0.17**	0.59	**0.15**
SNLI (N)	**0.16**	0.14	0.62	0.08
MultiNLI (All)	0.24	0.26	0.36	0.14
MultiNLI (E)	0.24	0.26	0.36	0.14
MultiNLI (C)	0.24	0.26	0.36	0.14
MultiNLI (N)	0.25	0.25	0.36	0.14
SQuAD (All)	0.15	0.09	0.61	0.15
SQuAD (+)	0.09	0.08	**0.71**	0.12
SQuAD (-)	**0.17**	**0.09**	0.58	**0.16**

Table 7: Attention weight of each attention function.

5.5 Discussion

To analyze the effectiveness of each attention function, we conduct ablation tests on four datasets by using single attention function and removing any of attention functions. Since the label of test data in MultiNLI is not published, we analyze it using the Matched and Mismatched data in the development set. As illustrated in Table 5, our MwAN is obviously better than using single attention function. In addition, removing any of attention functions leads to the worse score, which shows the effectiveness of our multiway attention network.

Next, we analyze the effect of four attention functions in different categories of samples. In our adaptive combinations, we can obtain the weight of each attention function at every position in P (Equation 9b). We average all weights of each attention function. Table 7 shows the result for four attention functions. We observe that each attention function has different effect on samples in different categories. The weight of the dot attention increases in positive samples (paraphrase in Quora, entailment in SNLI, and answer sentence in SQuAD) and the other functions increase in negative samples (non-paraphrase in Quora, contradiction in SNLI, and non-answer sentence in SQuAD), which indicates the dot attention is dominant in modeling similarity and other functions are good at modeling difference. Our multiway attention network combines their strengths, therefore achieves better results.

6 Conclusion

In this paper, we propose a multiway attention network that applies multiple attention functions to model the matching between a pair of sentences. We mainly focus on three tasks, namely paraphrase identification, natural language inference, and answer sentence selection. After obtaining contextual word representation based on the word embeddings. We propose using four different attention functions to match words in two sentences. Next, we aggregate the matching information from multiway attention functions. Finally, we use the attention pooling to aggregate the matching information into a fix-length vector and feed it into a classifier for the final decision. Experimental results show that our proposed multiway attention network outperforms baseline neural network models and yields state-of-the-art results on the Quora Question Pairs for the paraphrase identification, SNLI and MultiNLI for the natural language inference, and the answer sentence selection task on the SQuAD dataset.

Acknowledgments

We greatly thank Nan Yang for helpful discussions. We also thank all the anonymous reviewers for their helpful comments. Chuanqi Tan and Weifeng Lv are supported by the National Key R&D Program of China (No. 2017YFB1400200) and National Natural Science Foundation of China (No. 61421003 and 71501003).

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. In *EMNLP*, 2015.

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

sentence understanding. In *ACL*, pages 1466–1477. Association for Computational Linguistics, 2016.

[Chen *et al.*, 2016] Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the cnn/daily mail reading comprehension task. In *ACL*, pages 2358–2367. Association for Computational Linguistics, 2016.

[Chen *et al.*, 2017] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *ACL*, pages 1657–1668, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[Cheng *et al.*, 2016] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *EMNLP*, pages 551–561. Association for Computational Linguistics, 2016.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734. Association for Computational Linguistics, 2014.

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

[Madnani *et al.*, 2012] Nitin Madnani, Joel Tetreault, and Martin Chodorow. Re-examining machine translation metrics for paraphrase identification. In *NAACL*, pages 182–190. Association for Computational Linguistics, 2012.

[Manning *et al.*, 2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL System Demonstrations*, 2014.

[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 Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *EMNLP*. Association for Computational Linguistics, 2016.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543. Association for Computational Linguistics, 2014.

[Peters *et al.*, 2018] Matthew E Peters, Neumann Mark, Iyyer Mohit, Gardner Matt, Clark Christopher, Lee Kenton, and Zettlemoyer Luke. Deep contextualized word representations. *ICLR*, 2018.

[Rajpurkar *et al.*, 2016] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*. Association for Computational Linguistics, 2016.

[Rocktäschel *et al.*, 2015] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. Reasoning about entailment with neural attention. *CoRR*, 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. The COLING Organizing Committee, 2016.

[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*, 2014.

[Tan *et al.*, 2016] Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. Improved representation learning for question answer matching. In *ACL*, pages 464–473. Association for Computational Linguistics, 2016.

[Wang and Jiang, 2016a] Shuohang Wang and Jing Jiang. A compare-aggregate model for matching text sequences. In *ICLR*, 2016.

[Wang and Jiang, 2016b] Shuohang Wang and Jing Jiang. Learning natural language inference with LSTM. In *NAACL, San Diego California, USA, June 12-17*, 2016.

[Wang *et al.*, 2016a] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. Semi-supervised clustering for short text via deep representation learning. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 31–39. Association for Computational Linguistics, 2016.

[Wang *et al.*, 2016b] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. Sentence similarity learning by lexical decomposition and composition. In *COLING*, pages 1340–1349. The COLING Organizing Committee, 2016.

[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 *ACL*. Association for Computational Linguistics, 2017.

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

[Yang *et al.*, 2015] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *EMNLP*, pages 2013–2018, 2015.

[Yin *et al.*, 2016] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *TACL*, 4:259–272, 2016.

[Yu *et al.*, 2014] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. Deep learning for answer sentence selection. *Proceedings of the Deep Learning and Representation Learning Workshop: NIPS*, 2014.

[Zeiler, 2012] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.