The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

A Multi-View Fusion Neural Network for Answer Selection

Lei Sha,* Xiaodong Zhang,* Feng Qian, Baobao Chang, Zhifang Sui
* Contributed equally
Key Laboratory of Computational Linguistics, Ministry of Education
School of Electronics Engineering and Computer Science, Peking University
{shalei, zxdcs, nickqian, chbb, szf}@pku.edu.cn

Abstract

Community question answering aims at choosing the most appropriate answer for a given question, which is important in many NLP applications. Previous neural network-based methods consider several different aspects of information through calculating attentions. These different kinds of attentions are always simply summed up and can be seen as a "single view", causing severe information loss. To overcome this problem, we propose a Multi-View Fusion Neural Network, where each attention component generates a "view" of the QA pair and a fusion RNN integrates the generated views to form a more holistic representation. In this fusion RNN method, a filter gate collects important information of input and directly adds it to the output, which borrows the idea of residual networks. Experimental results on the WikiQA and SemEval-2016 CQA datasets demonstrate that our proposed model outperforms the state-of-the-art methods.

Introduction

In question answering (QA), answer selection aims to choose the most appropriate answer from a pre-selected candidate set (Yao et al. 2013), which could be important to various NLP tasks. In human-computer dialog systems, for example, the dialog agent needs to respond to questions issued by the user. Fortunately, a large number of QA pairs exist in community question answering (CQA) forums, such as Quora[1] and Stack Overflow.[2] It should be more convenient for a dialog agent to directly retrieve the most appropriate answer from these forums, rather than generating a new one by itself from scratch. Table 1 illustrates a simplified example for the answer selection task, where *Answer I* is less related to the query content "cause" and *Answer II* better matches the question.

Traditional methods are typically based on lexical and syntactic features, e.g., the edit distance between parse trees (Heilman and Smith 2010). Such methods require extensive human efforts on feature engineering. Recently, deep learning models have been adapted in many applications for natural language processing. Neural networks such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) can automatically learn features for answer selection. Attention mechanisms have also been introduced to enhance interaction between questions and answers (Wang, Liu, and Zhao 2016; Zhang et al. 2017).

Question	What causes heart disease?
Answer I	Cardiovascular disease refers to any disease that affects the cardiovascular system, principally cardiac disease, vascular diseases of the brain and kidney, and peripheral arterial disease. ($\times$)
Answer II	The causes of cardiovascular disease are diverse but atherosclerosis and/or hypertension are the most common. ($\surd$)

Table 1: An example question and candidate answers. (The second answer is correct.)

Previous attention-based models roughly collect all useful information from network hidden states and squeeze them to generate one attention vector (namely, a single view), which fails to capture various fine-grained aspects between the question and candidate answers.

Take the question in Table 1 for example. On one hand, as we analyze the question content, a good answer should contain the "cause" of heart disease. From another view to parse the question type, an ideal answer seems to be related to "what", rather than "who" or "where". A single attention vector, used in existing approaches, may not be sufficient to capture various different aspects of viewing a question, resulting in inaccurate answers.

In this paper, we propose a Multi-View Fusion Neural Network (MVFNN), which targets at better modeling on multiple question aspects like question type, question main verb, question semantics, etc. We focus four views in this study: inquiry type view, inquiry main verb view and inquiry semantic view, co-attention view. Then our model integrates all views into a holistic representation by a fusion recurrent network. In our proposed fusion RNN, we record the important parts of each time step and store them into an external memory, which is added to the output of the RNN. Based on the idea of deep residual networks (He et al. 2016), this

[1] https://www.quora.com/

[2] http://www.stackoverflow.com/

mechanism allows the original view information to directly affect the ultimate feature representation.

In summary, we make the following contributions in this paper:

- We consider the attention mechanism from several views to take different kinds of information into consideration.
- We propose a fusion RNN method to integrate different views together for a more holistic view of the QA pair. With the idea of deep residual network, the holistic view contributes to the performance a lot.
- Our proposed model achieves the state-of-the-art results on two datasets. We also conduct ablation tests to demonstrate the effect from each type of view as well as the fusion RNN method.

Related Work

Answer selection is usually formulated as a text matching problem. Previous work can be roughly divided into two branches: traditional feature-based methods and neural network-based methods. Traditional methods tend to employ feature engineering, linguistic tools, or external resources, which is time-consuming and suffers from the data sparsity problem. A bunch of related studies (Wang, Smith, and Mitamura 2007; Heilman and Smith 2010; Severyn and Moschitti 2013) utilize information on syntactic parse trees to match questions and answers. Xue, Jeon, and Croft (2008) combine a translation-based language model for the question part with a query likelihood approach for the answer part using a retrieval method. Surdeanu, Ciaramita, and Zaragoza (2011) analyze different kinds of feature types such as similarity features, translation features, density / frequency features for ranking answers to non-factoid questions in community QA forums. Yih et al. (2013) use semantic features from WordNet to enhance lexical features. Tymoshenko and Moschitti (2015) analyze the effect of syntactic and semantic structures extracted by shallow and deeper syntactic parsers in answer re-ranking. Filice et al. (2016), who achieved the best results in the SemEval-2016 CQA task, used various types of features including 8 similarity features, 44 heuristic features, and 16 thread-based features.

Recently, neural network based methods have achieved great progress in alleviating the burden of manual feature engineering. Deep Structured Semantic Models (DSSM) (Huang et al. 2013) and its convolutional counterpart C-DSSM (Shen et al. 2014) map the query and answers to a common semantic space using a non-linear projection, and calculate their relevance score by the cosine similarity between their semantic vectors. Lu and Li (2013) and Wang, Mi, and Ittycheriah (2016) use deep architectures to model the complicated matching relations between two texts. Iyyer et al. (2014) use recursive neural networks to add syntactic information (e.g. from dependency trees) to the model.

Wang and Nyberg (2015), Feng et al. (2015) and Hu et al. (2014) proposed to use a stacked bidirectional Long Short-Term Memory (LSTM) network to sequentially read sentences of question and answer, and then output their relevance scores. CNN based approaches are also proposed to model the two sentences directly on the interaction space to make them meet before their own high-level representations mature (Hu et al. 2014; Lingxun and Yan 2016; Tymoshenko, Bonadiman, and Moschitti 2016a; 2016b; Yin and Schütze 2017). Miao, Yu, and Blunsom (2015) proposed the neural variational inference method. With the development of attention mechanisms, Tan, Xiang, and Zhou (2015) and Wang, Liu, and Zhao (2016) propose inner attention and outer attention on the sentence RNNs of question and answer.

For better performance on redundant and noisy text, Zhang et al. (2017) propose an attentive tensor-based CNN to build models that are more robust to noisy texts.

The above attention-based studies tend to abstract all useful information together into a "single view". In contrast, we propose to calculate attention separately for each type of information in a "multi-view" fashion. Then we propose a fusion RNN to integrate the different views to form a more holistic view.

Attention from Four Views

For a give question, there could be a bunch of views to model its corresponding answer. In this paper, we construct four views according to our intuition. These four views are named inquiry type view, inquiry main verb view, inquiry semantic view and co-attention view. We still use the example question from Table 1 to illustrate the four views. Figure 1 illustrates the first three views and Figure 3 shows the fourth view.

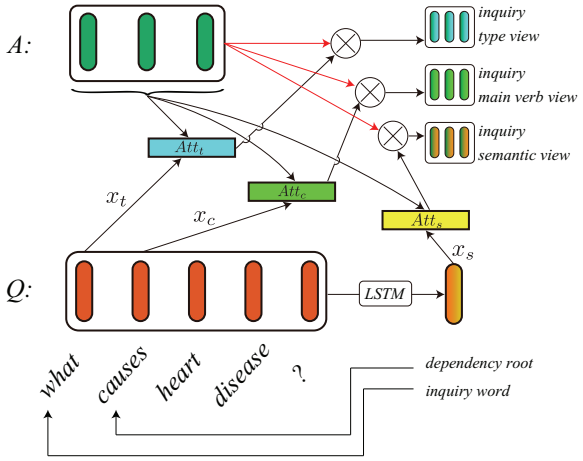

Figure 1: The generation process of inquiry type view, inquiry main verb view and inquiry semantic view. "⊗" represents element-by-vector multiplication.

Inquiry Type View

In this work, each word is represented using an embedding vector $x_i \in \mathbb{R}^d$. We denote the question as $X_Q = \{x_{q_1}, x_{q_2}, \cdots, x_{q_{|Q|}}\} \in \mathbb{R}^{d \times |Q|}$ and the answer $X_A = \{x_{a_1}, x_{a_2}, \cdots, x_{a_{|A|}}\} \in \mathbb{R}^{d \times |A|}$. $|Q|$ and $|A|$ represent the length of the question and answer, respectively.

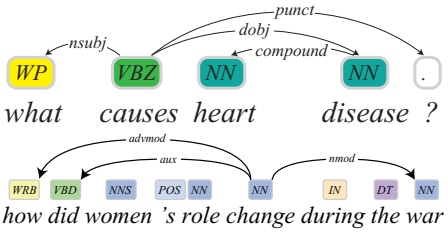

Figure 2: Two example dependency trees of questions, the root of the tree is the inquiry main verb.

For typical question sentences, for example those in the WikiQA dataset, the inquiry type could be determined by the interrogative word, since each of them contain exactly one interrogative. In Figure 1, the interrogative word is "what", illustrating the inquiry type. This word tells our model that the question requires an answer in the form of descriptions ("what") instead of proper names ("who" or "where").

We denote the interrogative as $x_t \in \mathbb{R}^d$, and then use the interrogative as well as the answer sentence to generate an attention Att_t:

$$Att_t = \text{softmax}(w_t^T \tanh(W_t x_t \oplus W_{ta} X_A)) \quad (1)$$

where $W_t, W_{ta} \in \mathbb{R}^{d\times d}$ and $w_t \in \mathbb{R}^d$ are parameters. "$\oplus$" means to add vector $W_t x_t$ to each row in the $W_{ta} X_A$ matrix.

Then, we can generate the inquiry type view as:

$$V_t = Att_t \otimes X_A, \quad V_t \in \mathbb{R}^{d\times|A|} \quad (2)$$

where "$\otimes$" means to multiply each element of Att_t and its corresponding vector in X_A.

Note that in Semeval-2016 CQA dataset, there is not necessarily an interrogative in each question. Instead, there is an inquiry type annotated for each word. So we just take this annotated type to calculate our inquiry type view.

Inquiry Main Verb View

Figure 2 shows two sampled question sentences represented in the form of dependency trees. In the first sentence, the root of the dependency tree, "cause", is the main verb of the whole question, we call it the *inquiry main verb*. Usually, the inquiry main verb tells us what the question is asking about and what it expects a system to answer. For example, the first question is asking about the reason and the second question expects an answer about "change". It is intuitive that utilizing the inquiry main verb could be helpful for answer selection. Note that finding the main verb in a sentence is a much less difficult problem compared with performing full dependency parsing. In this work we simply use a dependency parser only for the convenience of preliminary experimental study and leave more direct solutions of finding the main verb as future work.

We denote the inquiry main verb of the question as $x_c \in \mathbb{R}^d$, which is represented by a d-dimensional word vector. Then the attention Att_c can be calculated as:

$$Att_c = \text{softmax}(w_c^T \tanh(W_c x_c \oplus W_{ca} X_A)) \quad (3)$$

where $W_c, W_{ca} \in \mathbb{R}^{d\times d}$ and $w_c \in \mathbb{R}^d$ are trainable parameters.

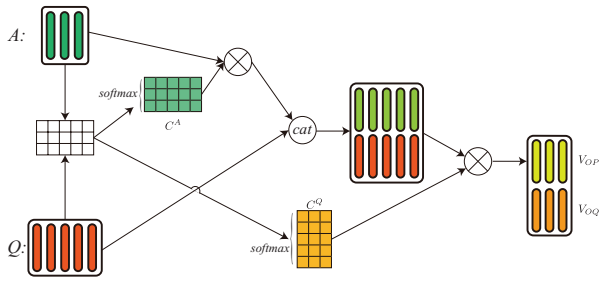

Figure 3: The generating process of co-attention view.

Inquiry main verb view is then obtained by:

$$V_c = Att_c \otimes X_A, \quad V_c \in \mathbb{R}^{d\times|A|} \quad (4)$$

Inquiry Semantic View

To understand the meaning of the whole question, we need to build the question's semantic information into the inquiry semantic view. We use an LSTM-RNN to read the question and take the average pooling of the output as the question's semantic information, given by:

$$x_s = Average\Big(LSTM(x_{q_1}, x_{q_2}, \cdots, x_{q_{|Q|}})\Big) \quad (5)$$

The equations for the inquiry semantic view are similar to previous ones.

$$\begin{aligned} Att_s &= \text{softmax}(w_s^T \tanh(W_s x_s \oplus W_{sa} X_A)) \\ V_s &= Att_s \otimes X_A, \quad V_s \in \mathbb{R}^{d\times|A|} \end{aligned} \quad (6)$$

Again, $W_s, W_{sa} \in \mathbb{R}^{d\times d}$ and $w_s \in \mathbb{R}^d$ are trainable parameters.

Co-attention View

Inspired by previous work on two-way attention from paired aspects (Santos et al. 2016; Xiong, Zhong, and Socher 2016), we also introduce a co-attention view in this work, focusing more on the interaction between the question and the answers. As is illustrated in Figure 3, we first compute the affinity matrix, which contains affinity scores that correspond to all pairs of question words and answer words: $M = X_A^T X_Q$. Then we normalize M row-wise and column-wise to obtain the attention weights C^Q and C^A:

$$\begin{aligned} C^Q &= softmax(M)^{\mathrm{T}} \in \mathbb{R}^{|Q|\times|A|} \\ C^A &= softmax(M^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{|A|\times|Q|} \end{aligned} \quad (7)$$

Therefore, we directly multiply X_A and C^A to obtain the summaries of the answer for each word in the question:

$$S_A = X_A C^A \in \mathbb{R}^{d\times|Q|} \quad (8)$$

Likewise, we compute the summary of the question and the summary of the previous attention contexts S_A in light of each word of the answer, then we get the co-attention QA pair view V_{CP} and co-attention question view V_{CQ}:

$$\begin{aligned} V_{OP} &= S_A C^Q \in \mathbb{R}^{d\times|A|} \\ V_{OQ} &= X_Q C^Q \in \mathbb{R}^{d\times|A|} \end{aligned} \quad (9)$$

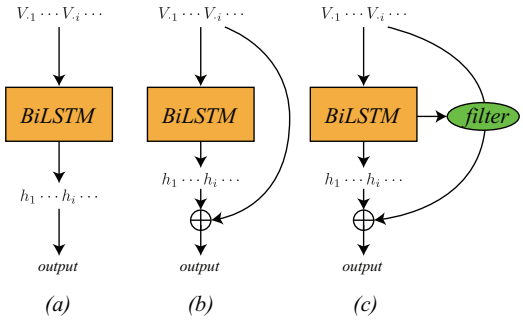

Figure 4: Differences of the three possible view-fusion methods: (a) Simple BiLSTM (b) Simple BiLSTM+ResNet (c) Fusion RNN.

We concatenate them to obtain the co-attention view: $V_O = [V_{OP}; V_{OQ}]$.

Finally, as is shown in Figure 5, we concatenate the inquiry type view V_t, inquiry main verb view V_c, inquiry semantic view V_s, and co-attention view V_O into: $V = [V_t; V_c; V_s; V_{OP}; V_{OQ}] \in \mathbb{R}^{5d \times |A|}$. V contains $|A|$ columns denoted as $V_{\cdot 1} \cdots V_{\cdot |A|}$, which represents the view of each word in the answer from the perspective of the QA pair.

Multi-view Fusion

Residual Connections

Residual networks (He et al. 2016) were proposed to add more direct links within deep neural nets, which makes the parameters easier to be optimized. Intuitively, selecting just important information in the form of residual link instead of passing all information could enhance the performance.

Inspired by residual networks, we propose to build a fusion recurrent network to integrate the aforementioned different views. In the fusion RNN, important original view information is added to the processed view information (which is the output of a bidirectional LSTM network) to form a holistic view. Therefore, the holistic view considers each view from both the original information and processed information. For comparison, we list other possible view-fusion methods in Figure 4.

Simple BiLSTM is the easiest way to fuse different views, which use a BiLSTM RNN to read the word-level view sequence $V = [V_{\cdot 1}, V_{\cdot 2}, \cdots, V_{\cdot |A|}]$, i.e.:

$$h_1, \cdots, h_{|A|} = BiLSTM(V_{\cdot 1}, \cdots, V_{\cdot |A|}) \tag{10}$$

Then the matching score could be calculated as:

$$\begin{aligned} h_{ave} &= Average(h_1, \cdots, h_{|A|}) \\ s(X_Q, X_A) &= w^\top h_{ave} \end{aligned} \tag{11}$$

where w is a parameter vector.

Simple BiLSTM+ResNet adds the merit of residual network to the simple BiLSTM model for comparison. The difference to simple BiLSTM is that the inputs of the BiLSTM are directly linked to the output:

$$\begin{aligned} v_{in} &= Average(V_{\cdot 1}, \cdots, V_{\cdot |A|}) \\ s(X_Q, X_A) &= w^\top (h_{ave} + W_{res} v_{in}) \end{aligned} \tag{12}$$

where the parameter matrix W_{res} transfers the dimension of v_{in} to match that of h_{ave}.

Fusion RNN is our proposed method. We take the BiLSTM as a filter to select important information from the input, then directly add it to the output as residual nets did.

Fusion RNN for Building Holistic View

As shown in Figure 4(c), we need to filter useful information from origin views. Intuitively, the important part of a view should be chosen in the word level. Therefore, we integrate each word level view (each column of V) to one vector using LSTM units, and then use the gate calculated by the word-level view to filter the important information needed to be remembered.

In order to select important information from each view, we sequentially read each column of V. The left of Figure 5 shows the $(t-1)$-th step of the forward building process. The forward building process is an LSTM-RNN architecture, in which the input $V_{\cdot (t-1)}$ is the $(t-1)$-th column of V. We keep a forward memory $M^{\rightarrow} \in \mathbb{R}^{d_M}$ across the building process. In each step, we need to integrate important information of the current answer word into the memory. For the $(t-1)$-th step, we calculate the information vector from a feed forward layer as:

$$I_{(t-1)} = W_i V_{\cdot (t-1)} + b_i \in \mathbb{R}^{d_M} \tag{13}$$

where $W_i \in \mathbb{R}^{d_M \times 5d}$ and $b_i \in \mathbb{R}^{d_M}$. Then, we take the output of the current LSTM unit $h^{\rightarrow}_{t-1} \in \mathbb{R}^{d_h}$ as a signal to tell us how much of $I_{(t-1)}$ should we fuse into the memory. We first calculate the gate variable z as:

$$z = \text{sigmoid}(W_h h^{\rightarrow}_{t-1}) \in \mathbb{R}^{d_M} \tag{14}$$

where $W_h \in \mathbb{R}^{d_M \times d_h}$. The sigmoid function can normalize the weight between 0 and 1. Then the holistic view memory of the t-th step can be updated by:

$$M^{\rightarrow}_t = (1 - z) \odot M^{\rightarrow}_{t-1} + z \odot I_{(t-1)} \tag{15}$$

where $\odot$ stands for element-wise multiplication.

Note that the current $M^{\rightarrow}_t$ contains the fusion result of the 1-st step to $(t-1)$-th time step, so we concatenate it to the input of the t-th step's LSTM unit for more accurate information selection of step t as is shown in Eq 16.

$$h^{\rightarrow}_t = \text{LSTMcell}([h^{\rightarrow}_{t-1}, M^{\rightarrow}_t]) \tag{16}$$

The backward process has the opposite direction to the forward process and is shown at the right part of Figure 5.

After the forward and backward important information collection processes, we concatenate $M^{\rightarrow}_{|A|}$ and $M^{\leftarrow}_{|A|}$ to get the final memory $M_{|A|}$; each time step's output of BiLSTM is also concatenated.

$$\begin{aligned} M_{|A|} &= [M^{\rightarrow}_{|A|}; M^{\leftarrow}_{|A|}] \\ h_t &= [h^{\rightarrow}_t; h^{\leftarrow}_t] \end{aligned} \tag{17}$$

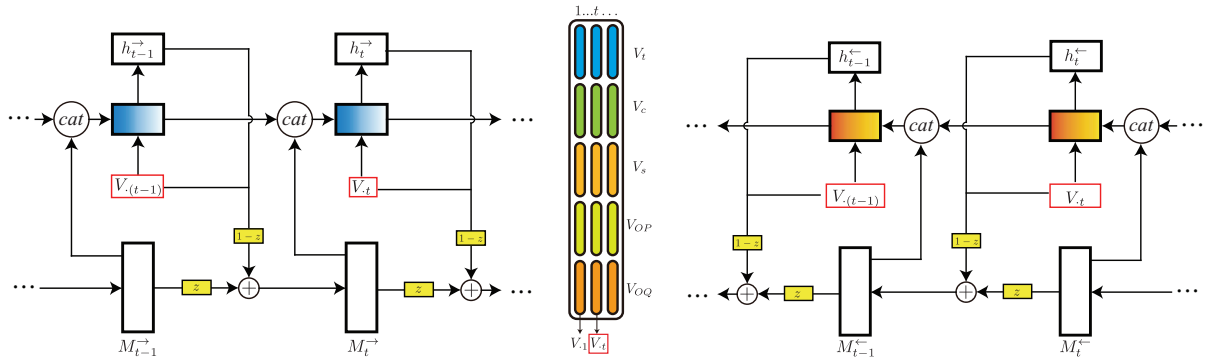

Figure 5: The bidirectional fusion RNN for building holistic view, the blue and orange rectangle are LSTM units. The matrix V in the middle is the concatenation of the views mentioned in Section . $V_{\cdot t}$ is the t-th column of V (bounded by red boxes), which is the input of the forward and backward LSTM.

Dataset (Train / Dev / Test)	WikiQA	SemEval-2016 CQA
# of questions	873 / 126 / 243	4879 / 244 / 327
# of answers	20360 / 2733 / 6165	36198 / 2440 / 3270
Ave length of question	7.16 / 7.23 / 7.26	43.29 / 46.88 / 49.77
Ave length of answer.	25.29 / 24.59 / 24.59	37.76 / 36.18 / 37.27

Table 2: The statistics of three answer selection datasets. For WikiQA, we remove all the questions that has no right answers.

Since the external memory $M_{|A|}$ keeps the important information of the input view $V_{\cdot 1} \cdots V_{\cdot |A|}$, we add $M_{|A|}$ to the average pooling of the BiLSTM's output as inspired by deep residual network (He et al. 2016).

$$\begin{aligned} h_{ave} &= Average(h_1, \cdots, h_{|A|}) \\ F &= h_{ave} + W_{mh} M_{|A|} \end{aligned} \tag{18}$$

where $W_{mh} \in \mathbb{R}^{2d_h \times 2d_M}$. Finally, F is the holistic view and is used for scoring the QA pair:

$$s(X_Q, X_A) = w^\top F \tag{19}$$

where $w \in \mathbb{R}^{2d_h}$.

Training

We use the max-margin criterion to train our model. For a given training QA pair (x_q, x_a^+) (which contains a question and the correct answer), its score should be larger than any other pairs containing the same question and a wrong answer (x_q, x_a^-) by a margin:

$$s(x_q, x_a^+, \theta) \leq s(x_q, x_a^-, \theta) + M \tag{20}$$

where M is a predefined margin; θ represents the parameter set.

In the training process, we take (x_q, x_a^+, x_a^-) as a training case and we denote the training set as Y. The objective function with hinge loss can be written as:

$$\begin{aligned} J(\theta) &= \frac{1}{|Y|} \sum_{(x_q, x_a^+, x_a^-) \in Y} \max\left\{0, l(x_q, x_a^+, x_a^-, \theta)\right\} \\ l(x_q, x_a^+, x_a^-, \theta) &= M + s(x_q, x_a^-, \theta) - s(x_q, x_a^+, \theta) \end{aligned} \tag{21}$$

To compute the network parameter θ, we maximize the max-margin likelihood $J(\theta)$ through stochastic gradient descent over shuffled mini-batches with the Adadelta (Zeiler 2012) update rule.

Experiments

Data & Preprocessing

We report the performance of our proposed method on two datasets: WikiQA (Yang, Yih, and Meek 2015) and SemEval-2016 CQA (Nakov et al. 2016). Table 2 demonstrates some statistics of the two datasets.

WikiQA is an open-domain question answering dataset in which each question has some candidate answers tagged as positive or negative and all answers are collected from Wikipedia. For each QA pair, we randomly sample a negative answer from the answer pool, and obtain a triple for training (Rao, He, and Lin 2016). As a result, we get a training set of 50,594 triplets.

SemEval-2016 CQA contains real data from the community-created Qatar Living Forums[3]. We focus on Subtask A: *Question-Comment Similarity*, where a comment is considered as an answer. Each comment is tagged with one of the three labels: "Good", "PotentiallyUseful", and "Bad". Only "Good" is considered as positive examples in evaluation. Unlike the WikiQA dataset, CQA have two main characteristics: **redundant** (questions and answers in CQA generally contains many sentences. Some of the sentences are meaningless and do not provide any useful information) and **noisy** (There are varies of informal language usage, abbreviations, typos and grammatical errors).

We use pre-trained GloVe (Pennington, Socher, and Manning 2014) word vectors[4] to initialize all word embeddings in our model. The word embeddings are also updated during training. All the dependency parse trees are produced by Stanford Corenlp[5].

[3] https://www.qatarliving.com/forum
[4] http://nlp.stanford.edu/projects/glove/
[5] http://stanfordnlp.github.io/CoreNLP/

Method	MAP	MRR
Yang, Yih, and Meek (2015)	0.6520	0.6652
Yin et al. (2015)	0.6921	0.7108
Miao, Yu, and Blunsom (2015)	0.6886	0.7069
Santos et al. (2016)	0.6886	0.6957
Wang, Mi, and Ittycheriah (2016)	0.7058	0.7226
He and Lin (2016)	0.7090	0.7234
Wang, Liu, and Zhao (2016)	0.7341	0.7418
Wang and Jiang (2016)	**0.7433**	**0.7545**
MVFNN	**0.7462**	**0.7576**

Table 3: Performances on WikiQA.

We use 100-dim word embeddings ($d = 100$) and we set the hidden layer length $d_h = 500$. The external memory length d_M is set to 400. The margin is set to 0.1.

Overall Performance

For evaluation, we use two ranking metrics: Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR). We evaluate our model on two datasets: WikiQA and SemEval-2016 CQA.

WikiQA The performance of WikiQA is shown in Table 3. The approaches listed in Table 3 are all attention-based methods except for Yang, Yih, and Meek (2015), which first proposed WikiQA task and reimplemented several baseline models. Among the attention-based methods, Miao, Yu, and Blunsom (2015) and Wang, Liu, and Zhao (2016) used simple semantic attention (the same as our inquiry semantic view). Yin et al. (2015), Wang and Jiang (2016) and Wang, Mi, and Ittycheriah (2016) introduced the attention mechanism into the CNN model, and captured the word-by-word similarity information. The method used by Santos et al. (2016) and He and Lin (2016) is similar to the co-attention view in our method. However, none of them ever used the inquiry content attention and most of them only used one kind of attention. Different from them, after fusing four different attention views, our model performs more effective than the other models as is shown in the last row of Table 3. We use student t-test to measure whether our method is significantly higher than previous methods. All the p-values turned out to be lower than 0.05.

SemEval-2016 CQA The performance of SemEval-2016 CQA is shown in Table 4. Hu et al. (2014) proposed a 2D deep convolution method. The multi-layer convolution and pooling operation may give it some advantage over Santos et al. (2016)'s simple co-attention method. Zhang et al. (2017) used element-to-element multiplication to model the word-by-word interaction which brings some of the original word meaning into the final representation to alleviate the influence of redundant and noisy.

In our model, we use the sequentially fusing method to dynamically decide what kind of information should we consider when calculating the matching score. As a result, we can see that our proposed method has significantly outperformed the state-of-the-art neural network approach

Method	MAP	MRR
Hu et al. (2014)	0.7798	-
Santos et al. (2016)	0.7712	-
Filice et al. (2016)	0.7919	**0.8642**
Joty et al. (2016)	0.7766	0.8493
Zhang et al. (2017) (w/o features)	0.7917	0.8311
Zhang et al. (2017)	**0.8014**	0.8423
MVFNN	**0.8005**	**0.8678**

Table 4: Experiment result in SemEval-2016. Since we did not use the additional features in Zhang et al. (2017), we only compare with their performance without additional features.

(without additional features) (Zhang et al. 2017) as well as the feature engineering approach (Filice et al. 2016). Student t test showed that all results are significant, and the p values are also listed in Table 4.

Effect of Multi-View

In this section, we will use experiments to illustrate the effect of multi-view. In Table 5, we listed single-view as comparison to multi-view.

Single-view is widely used in previous works. Single-view means to use all kinds of information (such as inquiry type, inquiry semantic, etc) to generate one view. The attention of single-view is calculated as Eq 22.

$$Att_{sing} = w^T((W_t x_t + W_c x_c + W_s x_s) \oplus W_a X_A) \quad (22)$$

where w, W_a, W_t, W_c, W_s are trainable parameters. An equivalent form of single-view is to use a feed forward neural network (denoted as a) to compute the attention as follows:

$$\begin{aligned} \alpha_i &= a([X_{Ai}; x_t; x_c; x_s]) \\ Att_{sing} &= \text{softmax}([\alpha_1, \cdots, \alpha_i, \cdots, \alpha_{|A|}]) \end{aligned} \quad (23)$$

where X_{Ai} represents the word vector of the i-th word in answer. Then the single-view is:

$$V_{sing} = Att_{sing} \otimes X_A \quad (24)$$

Since co-attention view V_O is created by 2-dim attention, we just concatenate them as $V = [V_{sing}, V_O]$. The fusion process is the same as MVFNN.

Also, several ablation experiments are listed in Table 5. We exclude the four kind of views from MVFNN one by one and report the results in Table 5.

From the results, we can see that all kinds of view bring more or less improvement to our model. Among them, the inquiry main verb view brings about 2% increment in MAP and MRR of the two tasks, and the inquiry semantic view brings 1%$\sim$2% increment. Note that co-attention view brings about 5%$\sim$6% increment on the two tasks, which means that co-attention view is extremely important in the QA matching process.

In the single-view experiments, we can see that our model of multi-view method MVFNN can significantly outperform

Method	WikiQA		SemEval-2016 CQA	
	MAP	MRR	MAP	MRR
Single-view	0.6882	0.7004	0.7780	0.8591
p-value	0.0015$*$	0.0016$*$	0.0122$*$	0.0256$*$
MVFNN	**0.7462**	**0.7576**	**0.8005**	**0.8718**
MVFNN $-$ Inquiry Type View	0.7368	0.7390	0.7851	0.8463
MVFNN $-$ Inquiry Main Verb View	0.7319	0.7411	0.7843	0.8499
MVFNN $-$ Inquiry Semantic View	0.7382	0.7576	0.7826	0.8575
MVFNN $-$ Co-attention View	0.7018	0.7130	0.7503	0.8238

Table 5: Comparison between single-view, multi-view and ablation experiments of separated attention. The p-value in the second line is calculated between "Single-view" and "MVFNN". "$*$" means significant.

Method	WikiQA		SemEval CQA	
	MAP	MRR	MAP	MRR
Simple BiLSTM	0.7253	0.7378	0.7855	0.8539
+ ResNet	0.7312	0.7479	0.7911	0.8644
Fusion RNN	**0.7462**	**0.7576**	**0.8005**	**0.8718**

Table 6: Comparison between simple BiLSTM and our method.

the single-view method. This fact states that simply add useful information to generate attentions will lead to severe information loss. So we just separately calculate the attentions and let the fusion process to decide which kind of information is important. Therefore, we get an improvement in the result.

Effect of Fusion RNN

According to Table 6, we can see that after adding residual connections, the model outperforms simple BiLSTM model. Also, our fusion RNN method outperforms the simple BiLSTM and "simple BiLSTM+ResNet" due to the external memory. As is illustrated in Figure 4, simple BiLSTM only takes the $V_{\cdot 1}, \cdots, V_{\cdot |A|}$ as input and calculates $h_1, \cdots, h_{|A|}$ as output. After adding ResNet, the output layer can directly take some original information, which makes the model be trained with reference to the layer inputs. Unlike the ResNet, our fusion RNN model use the hidden layer of the BiLSTM to calculate a filter gate to keep the important part of the input. Therefore, our model can be trained with the reference to the important part of the inputs. This improvement in architecture leads to a better performance in result.

Conclusion

In this paper, we propose a multi-view fusion method to solve answer selection problem. We proposed to separate various kind of attention to form different views, including inquiry type view, inquiry main verb view, inquiry semantic view and co-attention view. Then we use a fusion RNN to integrate these views to get a holistic view of the QA pair, which can be treated as an adaptation of residual networks in our scenario. We did experiments on two datasets: WikiQA and SemEval-2016 CQA, and we achieved state-of-the-art performance on the two datasets. Also, we did detailed ablation experiments to verify the effectiveness of our proposed components.

Future studies may focus on the exploitation of heterogeneous resources and the integration of multiple tasks. Conceptually related tasks may include sentence paraphrasing, relation extraction, entity classification and linking. It could be helpful to jointly train answer selection models with any of these tasks. Deriving additional attention views from these tasks may also contribute to performance gain. Furthermore, since unlabeled data are much easier to acquire, we also would like to explore whether introducing additional unlabeled data or weak supervision could be helpful for calculating additional attention vectors.

Acknowledgments

We thank Jin-ge Yao for helpful discussion of this paper. We also thank reviewers for helpful advice. The research work is supported by the National Key Research and Development Program of China under Grant No. 2017YFB1002101 and project 61772040 supported by NSFC. The contact author is Zhifang Sui.

References

Feng, M.; Xiang, B.; Glass, M. R.; Wang, L.; and Zhou, B. 2015. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 813–820. IEEE.

Filice, S.; Croce, D.; Moschitti, A.; and Basili, R. 2016. Kelp at semeval-2016 task 3: Learning semantic relations between questions and answers. *Proceedings of SemEval* 16:1116–1123.

He, H., and Lin, J. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of NAACL-HLT*, 937–948.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Heilman, M., and Smith, N. A. 2010. 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*, 1011–1019. Association for Computational Linguistics.

Hu, B.; Lu, Z.; Li, H.; and Chen, Q. 2014. Convolutional neural network architectures for matching natural language sen-

tences. In *Advances in Neural Information Processing Systems*, 2042–2050.

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2333–2338. ACM.

Iyyer, M.; Boyd-Graber, J. L.; Claudino, L. M. B.; Socher, R.; and Daumé III, H. 2014. A neural network for factoid question answering over paragraphs. In *EMNLP*, 633–644.

Joty, S.; Moschitti, A.; Al Obaidli, F. A.; Romeo, S.; Tymoshenko, K.; and Uva, A. 2016. Convkn at semeval-2016 task 3: Answer and question selection for question answering on arabic and english fora. *Proceedings of SemEval* 896–903.

Lingxun, M., and Yan, L. 2016. m^2 s-net: Multi-modal similarity metric learning based deep convolutional network for answer selection. *arXiv preprint arXiv:1604.05519*.

Lu, Z., and Li, H. 2013. A deep architecture for matching short texts. In *Advances in Neural Information Processing Systems*, 1367–1375.

Miao, Y.; Yu, L.; and Blunsom, P. 2015. Neural variational inference for text processing. *arXiv preprint arXiv:1511.06038*.

Nakov, P.; Màrquez, L.; Moschitti, A.; Magdy, W.; Mubarak, H.; Freihat, A. A.; Glass, J.; and Randeree, B. 2016. Semeval-2016 task 3: Community question answering. *Proceedings of SemEval* 16.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Rao, J.; He, H.; and Lin, J. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 1913–1916. ACM.

Santos, C. d.; Tan, M.; Xiang, B.; and Zhou, B. 2016. Attentive pooling networks. *arXiv preprint arXiv:1602.03609*.

Severyn, A., and Moschitti, A. 2013. Automatic feature engineering for answer selection and extraction. In *EMNLP*, 458–467.

Shen, Y.; He, X.; Gao, J.; Deng, L.; and Mesnil, G. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*, 373–374. ACM.

Surdeanu, M.; Ciaramita, M.; and Zaragoza, H. 2011. Learning to rank answers to non-factoid questions from web collections. *Computational Linguistics* 37(2):351–383.

Tan, M.; Xiang, B.; and Zhou, B. 2015. Lstm-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.

Tymoshenko, K., and Moschitti, A. 2015. Assessing the impact of syntactic and semantic structures for answer passages reranking. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1451–1460. ACM.

Tymoshenko, K.; Bonadiman, D.; and Moschitti, A. 2016a. Convolutional neural networks vs. convolution kernels: Feature engineering for answer sentence reranking. In *Proceedings of NAACL-HLT*, 1268–1278.

Tymoshenko, K.; Bonadiman, D.; and Moschitti, A. 2016b. Learning to rank non-factoid answers: Comment selection in web forums. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2049–2052. ACM.

Wang, S., and Jiang, J. 2016. A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:1611.01747*.

Wang, D., and Nyberg, E. 2015. A long short-term memory model for answer sentence selection in question answering. *ACL, July*.

Wang, B.; Liu, K.; and Zhao, J. 2016. Inner attention based recurrent neural networks for answer selection. In *The Annual Meeting of the Association for Computational Linguistics*.

Wang, Z.; Mi, H.; and Ittycheriah, A. 2016. Sentence similarity learning by lexical decomposition and composition. *arXiv preprint arXiv:1602.07019*.

Wang, M.; Smith, N. A.; and Mitamura, T. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa. In *EMNLP-CoNLL*, volume 7, 22–32.

Xiong, C.; Zhong, V.; and Socher, R. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.

Xue, X.; Jeon, J.; and Croft, W. B. 2008. Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, 475–482. ACM.

Yang, Y.; Yih, W.-t.; and Meek, C. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of EMNLP*, 2013–2018. Citeseer.

Yao, X.; Durme, B. V.; Callison-Burch, C.; and Clark, P. 2013. Answer extraction as sequence tagging with tree edit distance. In *HLT-NAACL*.

Yih, W.-t.; Chang, M.-W.; Meek, C.; and Pastusiak, A. 2013. Question answering using enhanced lexical semantic models.

Yin, W., and Schütze, H. 2017. Task-specific attentive pooling of phrase alignments contributes to sentence matching. *arXiv preprint arXiv:1701.02149*.

Yin, W.; Schütze, H.; Xiang, B.; and Zhou, B. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zhang, X.; Li, S.; Sha, L.; and Wang, H. 2017. Attentive interactive neural networks for answer selection in community question answering. In *Thirty-first AAAI Conference on Artificial Intelligence*.