

# Exploring Answer Stance Detection with Recurrent Conditional Attention

Jianhua Yuan,<sup>1</sup> Yanyan Zhao,<sup>2</sup> Jingfang Xu,<sup>3</sup> Bing Qin<sup>1\*</sup>

<sup>1</sup>Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China

<sup>2</sup>Department of Media Technology and Art, Harbin Institute of Technology, China

<sup>3</sup>Sogou Technology Inc, Beijing, China

{jhyuan, yyzhao, qinb}@ir.hit.edu.cn, xujingfang@sogou-inc.com

## Abstract

Detecting stance from certain types of question-answer pairs is an interesting problem which has not been carefully explored. Unlike previous stance detection tasks, targets here are not given entities or claims but entire questions, which makes it difficult to capture the semantics of targets and build target-dependent representations of answers. To address them, we introduce the Recurrent Conditional Attention (RCA) model which incorporates a conditional attention structure into the recurrent reading process. RCA iteratively guides the distillation of question semantic with answer information and collects stance-oriented text relating to question, further revealing mutual relationship among stance, answer and question. Experiments on a manually labeled Chinese community QA stance dataset show that RCA outperforms four strong baselines by average 2.90% on *macro-F1* and 2.66% on *micro-F1* respectively.

## Introduction

The goal of stance detection is to classify the attitude expressed in a text towards a given target, as *favor*, *against*, or *neutral*. Most previous work focused on stance detection problems in corpus including online debates (Hasan and Ng 2013; Sridhar et al. 2015), news articles (Ferreira and Vlachos 2016), tweets (Mohammad et al. 2016), fake news (Bhatt et al. 2018), and rumors (Qazvinian et al. 2011). To our best knowledge, few has noticed that in certain types of QA pairs, answerers usually express their opinions about/towards entities or claims hold by the questions. Mining these QA pairs and detecting people’s stances embedded in them will benefit a variety of tasks including opinion-oriented question answering and opinion summarization of QA pairs.

In this paper, we focus on the problem of answer stance detection in community QA (**AnswerStance**). As shown in Figure 1, given a question and an answer, the task aims to determine whether the answer holds a *favor*, *against* or *neutral* attitude towards the entity or claim in that question. Apart from Yes/No type questions in QA tasks, we also deal with questions asking for viewpoint, such as the example 2 in Figure 1. Moreover, this task differs from traditional stance

Example 1	
<b>Question</b>	打雷能用电脑吗? Can I use computer during thundering ?
<b>Answer</b>	如果你想换电脑...那么就用吧...旧的不去新的不来 If you want to have a new one, then just use it ... New one won't come until the old go.
<b>Stance</b>	<i>Against</i>
Example 2	
<b>Question</b>	新的ipad 2018 怎么样? How about the new ipad 2018 ?
<b>Answer</b>	价格便宜, 居然还支持 apple pencil The price is low, it even supports apple pencil
<b>Stance</b>	<i>Favor</i>

Figure 1: Two types of example from answer stance detection task.

detection task, e.g. stance classification in tweets and fake news, in that: targets here are not given entities or claims but the whole questions. Such difference requires abstracting the semantic meanings of questions before inferring the stance.

Various models have been proposed to solve stance detection problems. Some apply traditional machine learning models with manually collected rich features (Sobhani, Mohammad, and Kiritchenko 2016). Others implement deep learning models such as Recurrent Neural Networks (RNNs) (Augenstein et al. 2016; Du et al. 2017) and Convolutional Neural Networks (CNNs) (Wei et al. 2016; Vijayaraghavan et al. 2016). While these methods deal with given targets including either entities like “Donald Trump” or entire claims like tweets and news, none has tackled targets in question forms. More importantly, in AnswerStance task, semantic representations of the question (target) should be further improved with the answer (text) information in order to get better understand of the stance, which so far has been neglected by most of these methods.

Therefore, we introduce the recurrent conditional attention (RCA) network, which repeatedly employs the conditional attention structure to model the relationship be-

\*Corresponding Author

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

tween question and answer. At each reading step, RCA conditions the representation of answer text on most recently updated question vector. Between two consecutive steps, the latest answer representation guides the distillation of question vector. Through the above reading process, RCA gradually polishes the higher level stance state for final prediction. We then further evaluate our model on a manually annotated Chinese QA stance dataset, which is publicly available along with the source code at <https://github.com/surpriseshef/AnswerStance>.

In summary, the main contributions of this paper are three-fold:

- We propose the task of *answer stance detection* and create a Chinese QA stance dataset comprised of over **13,000** pairs with *favor*, *against* and *neutral* stance labels.
- We develop the RCA model to leverage the mutual dependency between questions and answers through a recurrent conditional attention structure.
- We conduct extensive experiments on the stance dataset. Empirical results reveal that the RCA model is superior to a number of baselines and several competitive systems for stance detection tasks.

## Problem Formulation

In this section, we introduce the answer stance classification problem and formulate it as a target-dependent task.

### Notation and Problem Definition

Formally, given a question sentence of  $N$  words  $X^Q = \{x_1^Q, x_2^Q, \dots, x_N^Q\}$  and an answer of  $M$  words  $X^A = \{x_1^A, x_2^A, \dots, x_M^A\}$ , where each  $x_i$  is a word represented by a  $d$ -dimension vector from a large pre-trained word embedding matrix, the task of answer stance detection in community question-answering (AnswerStance) aims at determining the stance label of answer text  $X^A$  towards the question  $X^Q$ . For the Example 2 in Figure 1, the answer explicitly expresses a *favor* stance by praising the price of “new ipad 2018”.

### Comparison with Other Tasks

Different from Yes/No question answering task (Nakov et al. 2015), we also tackle viewpoint type questions. In Yes/No type instances, answers can be seen as conveying authors’ *favor* or *against* stance towards statements made in questions. In viewpoint type, a questioner usually asks for overall opinions of an entity.

Unlike previous stance detection tasks, where either entities like “Donald Trump”(Mohammad et al. 2016) or claims like tweets (Derczynski et al. 2017) and news headlines (Bhatt et al. 2018) are given as targets, in AnswerStance, such kind of targets are not given but replaced by entire questions. In fact, the actual targets in our task manifest as entities in viewpoint questions and claims in Yes/No questions. While treating the entire question as a target doesn’t introduce ambiguity, it still retains all the semantic meaning of the entity or claim in that question.

## Approach

In this section, we introduce the Recurrent Conditional Attention model (RCA) for answer stance detection in cQA. We first give an overview of the approach. Then we present how to polish stance states with conditional attention structure over QA pairs. Finally, we describe the training objectives. An overall architecture of RCA model is shown in Figure 2.

### An Overview of RCA

Given a question  $X^Q$  and an answer  $X^A$ , we first map each word into its embedding vector. Since further understanding of questions requires more powerful models than simple bag-of-words (BOW) commonly adopted in previous work (Tang, Qin, and Liu 2016; Du et al. 2017; Chen et al. 2017), we turn to recurrent neural networks (Elman 1990) for better semantic representations of words in context. In practice, we adopt Gated Recurrent Unit network (GRU) (Cho et al. 2014) in our model. Given an input sequence  $X = \{x_1, x_2, \dots, x_t, \dots, x_T\}$  of  $T$  tokens, at each time step  $t$ , a GRU updates its hidden state  $h_t$  according to the current input  $x_t$  as well as the previous hidden state  $h_{t-1}$ :

$$h_t = GRU(h_{t-1}, x_t) \quad (1)$$

We employ two bi-directional GRU (Bi-GRU) to retain context from both sides for questions and answers. For the  $i$ -th word, the concatenation of bi-directional hidden states of Bi-GRU is used, which namely are  $h_i^Q$  for question word and  $h_i^A$  for answer word.

Motivated by the idea of *attention-over-attention* (Cui et al. 2017), we would like to build an answer-aware question vector as well as a question-dependent answer representation. To achieve these, we incorporate a conditional attention structure into the recurrent reading process. Specifically, after we learned the contextual vectors of each word in both questions and answers, we apply another GRU based module to conditionally attend these representations and update the stance state accordingly.

We use  $s_t$  to denote stance state that summarize the stance information after reading the QA pair for  $t$  times and start with  $s_0$  set to all  $\mathbf{0}$  in order to mimic *neutral* stance before reading any particular QA pair. At the first reading step, we regard  $s_0$  as the input to adaptively select important evidences  $\tilde{Q}_1$  from question through attention. We update stance state  $s$  for one GRU step with  $s_0$  and  $\tilde{Q}_1$ . The temporal stance state  $s_1^*$ , combining both previous stance clues and most recently selected question context, is fed to attend state-related information  $\tilde{A}_1$  from answer hidden states. After that, we output the stance representation  $s_1$  for current reading step by fusing  $s_1^*$  and  $\tilde{A}_1$  with another GRU step. In a similar way, we run the above conditional attention multiple times, so that more abstractive evidence from question and answer could be accumulated to the stance state for final prediction.

The stance state of last reading step is considered as the representation of answer with regard to the target, and is further used as the feature for stance classification.

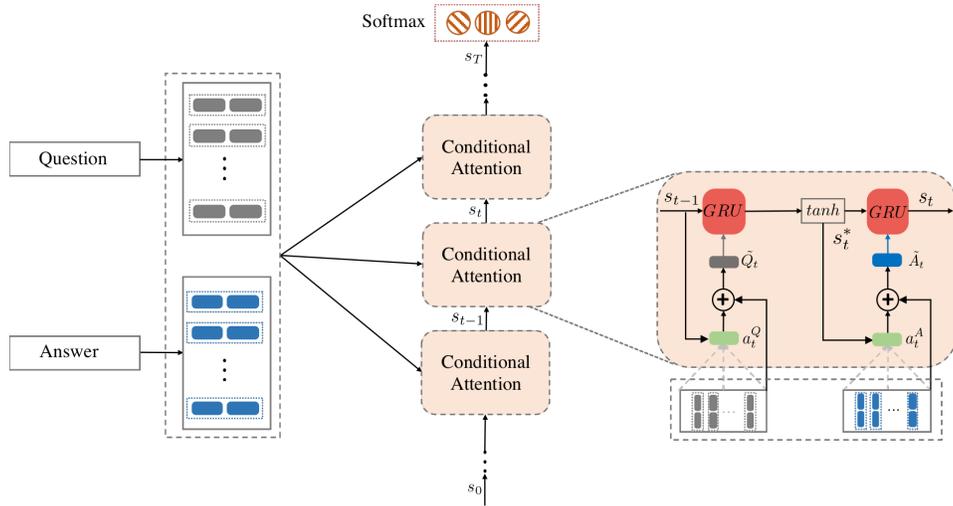


Figure 2: An overall architecture of RCA.

### Conditional Attention

We describe details of conditional attention in this part. The intuition is that mining the dependent relationship between question and answer would be useful to find stance-oriented evidence from both sides. Hence, the representations of question and answer should be conditioned on each other. In one conditional attention structure, the representation of answer text is attended by the stance state with most recently updated question vector. Between two consecutive conditional attention structures, the previous step’s stance state with latest added answer information helps capture the semantic meaning of a question.

In the following, we present how RCA distills targets semantics from questions and builds target-dependent answer representations with respect to previously accumulated stance state  $s_{t-1}$ .

**Question Attention** The basic idea is that different words in question should be assigned distinct importance during capturing the semantics of questions. Words like “How about” are not likely to provide informative clues of a question while words like “German team” usually appear as an entity of stance receiver. Thus, we introduce attention mechanism (Bahdanau, Cho, and Bengio 2014) to guide the discovery of those importance words. We aggregate the semantic meaning of a question  $Q$  with respect to previous stance state  $s_{t-1}$  that describes the relationship among question, answer and stance.

We use the inner product between previous stance state  $s_{t-1}$  and hidden states of question words to score the contribution of each word made to question representations at reading step  $t$  as follows:

$$u_{ti}^Q = s_{t-1}^T h_i^Q \quad (2)$$

where  $h_i^Q$  is the  $i$ -th hidden state concatenation of an answer,  $u_{ti}^Q$  is the unnormalized attention weight of  $i$ -th word in question at reading step  $t$ .

Then, we normalize the importance score  $u_{ti}^Q$  through a softmax layer to get attention weight  $a_{ti}^Q$  of each question word:

$$a_{ti}^Q = \frac{\exp(u_{ti}^Q)}{\sum_1^N \exp(u_{tj}^Q)} \quad (3)$$

where  $N$  is the number of words in the question sentences. After that, we compute target representation  $\tilde{Q}_t$  at reading step  $t$  as a weighted sum of all hidden states of a question by following equation:

$$\tilde{Q}_t = \sum_1^N a_{ti}^Q * h_i^Q \quad (4)$$

**Answer Attention** Similarly, not all words in answer sentence contribute equally to the final stance toward the question, words like “surely” and “bad” play a more important role in stance prediction while other words like “today” and “balloon” are less likely to imply attitudes.

After we acquire the new question vector  $\tilde{Q}_t$ , we hope to build a target-dependent answer representation conditioned on previous stance state  $s_{t-1}$  and question vector  $\tilde{Q}_t$ . We then update  $s_{t-1}$  with  $\tilde{Q}_t$  using a GRU step and a  $\tanh$  activation<sup>1</sup>:

$$s_t^* = \tanh(GRU_{CA}(s_{t-1}, \tilde{Q}_t)) \quad (5)$$

Ideally,  $s_t^*$  should contain more recent question semantics together with the relationship among answer, question and stance. We measure the importance of each answer word and acquire a weighted sum vector  $\tilde{A}_t$  conditioned on temporal stance state  $s_t^*$  in similar ways:

$$u_{ti}^A = (s_t^*)^T h_i^A \quad (6)$$

$$a_{ti}^A = \frac{\exp(u_{ti}^A)}{\sum_1^M \exp(u_{tj}^A)} \quad (7)$$

<sup>1</sup>In preliminary experiments, we tried directly using GRU outputs without a nonlinear transformation, and found that adding a  $\tanh$  layer works slightly better.

$$\tilde{A}_t = \sum_1^M a_{ti}^A * h_i^A \quad (8)$$

where  $h_i^A$  is the representation of  $i$ -th word from an answer and  $M$  is the number of words in an answer.  $\tilde{A}_t$  stands for the new answer vector for polishing the stance state.

### Recurrent Stance Polishing

As mentioned earlier, the stance for current reading step is obtained from both the history stance vector  $s_{t-1}$  and new context  $\tilde{Q}_t$  and  $\tilde{A}_t$  of QA pair. Obviously, the final states for drawing stance labels should be directly conditioned on  $\tilde{A}_t$ . Therefore, we update stance states with  $\tilde{A}_t$  using another GRU step:

$$s_t = GRU_{CA}(s_{t-1}^*, \tilde{A}_t) \quad (9)$$

By this time, we finish one reasoning step over the QA pairs. Note that, during one reading step, we take two GRU time steps sequentially. Following previous works in target dependent sentiment analysis (Tang, Qin, and Liu 2016; Chen et al. 2017), we repeat the reading process for  $k$  times and polish stance states with conditional attention structure, where  $k$  is an empirically chosen small number such as 3 and 5.

When reading for multiple steps, mutual conditional relationships between questions and answers are introduced naturally, which greatly helps distill the semantic meanings of questions with answer information and vice versa. Such deliberation process ensures that RCA is capable of gradually retrieving stance-oriented semantic representations from question-answer pairs.

### Model Training

We treat the last stance state  $s_k$  of the reading process as final stance representation. We then feed  $s_k$  into a softmax layer to predict the possible stance label. Given the training dataset  $D$  and stance category set  $C$ , the RCA model is trained in an end-to-end fashion by minimizing the cross-entropy loss between gold labels and predictions together with an  $L2$  regularization term:

$$\mathcal{L} = - \sum_{i=1}^{|D|} \sum_{j=1}^{|C|} y_i^j \log f(x_i^Q, x_i^A) + \lambda \|\theta\|^2 \quad (10)$$

where  $|D|$  is the number of QA pairs in  $D$ ,  $i$  is the index of QA pair in dataset,  $j$  is the index of category,  $f(x_i^Q, x_i^A)$  is the predicted stance distribution of the model and  $\theta$  is the parameter set.

## Experimental Setup

In this section, we introduce statistics of our stance dataset in community QA, implementation details, evaluation metrics and comparative methods.

### Datasets

For our experiments, we collect question-answer pairs from popular Chinese question and answer websites including

Dataset	<i>against</i>	<i>neutral</i>	<i>favor</i>
Training	4050	1460	5088
Test	856	1018	1119

Table 1: Statistics of annotated answer stance dataset in community question-answer.

*Baidu Knows*<sup>2</sup>, *Sogou Wenwen*<sup>3</sup> and *Mingyi*<sup>4</sup>. We parse these websites with pre-defined templates to get structured question-answer pairs. Duplicate instances are removed before labeling. We ask three annotators to annotate the extracted question-answer pairs with most salient stance, choose the labels through majority vote and abandon pairs with ambiguous stances. The inter annotator agreement is 78.2%. While question-answer pairs without stances are mostly filtered with some rules based on URL and text, the remaining of them are removed during annotation process. Finally, we get a dataset of 13,591  $\langle question, answer \rangle$  pairs which are mainly relating to daily topics like pregnancy, food safety, diseases, etc. The detailed statistics of the dataset is shown in Table 1.

### Implementation details

All models are implemented using *PyTorch*<sup>5</sup>. A 100-dimensional word embeddings trained on a 10-billion-words Chinese encyclopedia corpus with *word2vec* (Mikolov et al. 2013) is used as the input of all models and is then fine-tuned during training. The weights of networks are initialized with a uniform distribution  $\mu(-0.1, 0.1)$ . We apply Adam (Kingma and Ba 2014) to optimize the models and initial learning rate is set to  $1e-4$ . We split one-tenth of the training set for tuning parameters and apply early-stopping according to performance on validation set during training. Mini-batch size is set to 8 for all models and dropout of 0.5 is adopted for preventing over-fitting. The maximum sequence lengths of question and answer are set to 25 and 45 respectively according to the statistics of training set. We segment all the Chinese sentences with LTP (Che, Li, and Liu 2010).

### Evaluation Metrics

Following previous evaluations (Mohammad et al. 2016; Xu et al. 2016) which treat *neutral* as a class not of interest, we also adopt both *macro-* and *micro-average F1* score of *favor* and *against* classes. However, different from SemEval 2016, we only apply common *macro-* and *micro-F1* score here instead of an arithmetic mean of  $F_{favor}$  and  $F_{against}$ . We present the *accuracy* of all labels for comparison with different baselines as well.

### Comparative Methods

We compare with the following baseline methods:

<sup>2</sup><https://zhidao.baidu.com/>

<sup>3</sup><http://wenwen.sogou.com/>

<sup>4</sup><http://ask.mingyi.com/>

<sup>5</sup><http://pytorch.org/>

Model	Acc	$F_{against}$	$F_{favor}$	$F_{macro}$	$F_{micro}$
BOW	0.5132	0.5779	0.6519	0.6157	0.6181
CNN	0.5359	0.6373	0.6422	0.6408	0.6402
LSTM	0.5316	0.6148	0.6541	0.6346	0.6367
BLSTM	0.5747	0.6336	0.6854	0.6599	0.6623
TAN	0.5780	0.6410	0.6917	0.6667	0.6692
RAM	0.5874	0.6742	0.6885	0.6815	0.6824
BiCond	0.5887	0.6623	0.6885	0.6754	0.6771
AoA	0.5864	0.6586	0.6963	0.6775	0.6796
<b>RCA</b>	<b>0.6204</b>	<b>0.7066</b>	<b>0.7016</b>	<b>0.7043</b>	<b>0.7037</b>

Table 2: Performance comparison of different stance detection models on test set.

- **BOW**: The Bag-of-Words model (BOW) sums all the word vectors in a sentence equally.
- **(B)LSTM**: We use two separate (B)LSTMs for question and answer respectively.
- **CNN**: Similarly, we use two separate CNNs for question and answer respectively.
- **TAN**: Du *et al.* (2017) applied Target-specific Attention model (TAN) to target-dependent stance classification in tweets and micro-blogs.
- **BiCond**: Augenstein *et al.* (2016) proposed Bi-directional Conditional Encoding model to solve stance detection problem in tweets.
- **AoA**: Cui *et al.* (2017) developed an *attention-over-attention* model to dynamically choose the appropriate part of text from answer document dependent on query with inter sentence attentions.
- **RAM**: Chen *et al.* (2017) adopts a memory network based structure which recurrently attends the text with BOW vectors of targets.

For BOW, LSTM, BLSTM and CNN models, we concatenate question and answer vectors as stance features. For TAN, BiCond and RAM methods, we re-implement them by following settings in original papers and choose the best hyper-parameters on development set. Besides, we let the RCA model read the QA pair for three times (*i.e.* 3 reading steps).

Since we try to detect stance from QA pairs with neural models rather than hand-craft features, we leave the comparison with feature-based SVM models in SemEval2015 Task 3 (Nakov et al. 2015). Instead, we take CNN and AoA models as representatives from QA community. Also, we compare with TAN and BiCond from previous stance detection tasks. Furthermore, RAM is adopted for comparison with Memory Network based model that has multiple attention layers.

## Results and Discussion

### Main Results

Experimental results are given in Table 2.

**Comparison of baselines.** As in many cases, the stance labels can be directly and correctly predicted from short answer sentence, even simple BOW models can easily reach

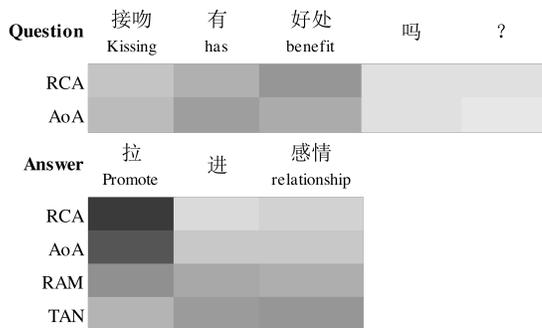


Figure 3: Visualization of attention weights from four models. Note that only RCA and AoA generate attention weights on questions and we only present attention weights at last reading step from RAM and RCA models. Deeper color implies larger probability.

*macro-* and *micro-F1* scores over 0.6. Since solving AnswerStance problems require understanding of answers with regard to questions, TAN, RAM, BiCond and AoA models with either attention or conditional encoding structures outperforms BOW, CNN and LSTM that simply concatenate question and answer vectors, by a large margin in terms of both F1 scores.

**Comparison with attention-based models.** AoA model outperforms TAN by over 1% in both *macro* and *micro-F1* scores for that AoA also utilizes the answer text for better question representation, which in return contributes to more accurate answer representations. Since answers usually express implicit stances with indirect expressions to the questions, semantic relationships between them are more complex to model. As a result, RAM which employs a recurrent attention module to reason over the QA pairs multiple times, achieves top performances among all the baseline models. We can observe that our model consistently outperforms all the baseline models in terms of both *macro* and *micro-F1* scores which reveals that combining both the conditional attention and polishing process works better at deliberating the real stance.

To verify the ability of RCA model to distill targets question, we also visualize the attention results for one QA pair. Note that only RCA and AoA are capable of outputting im-

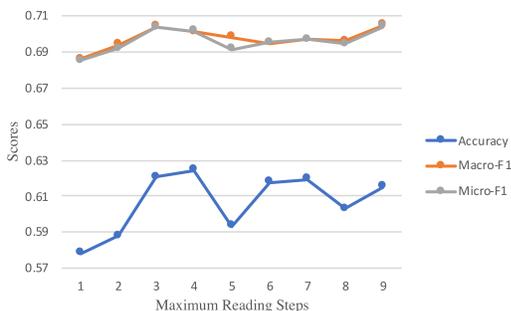


Figure 4: Performances of RCA models with different maximum reading steps. Evaluation metrics include *accuracy* of three class, *micro* and *macro F1* score of *favor* and *against* class.

Model	$F_{macro}$	$F_{micro}$
Full RCA model	<b>0.7043</b>	<b>0.7037</b>
– Question GRU	0.6772	0.6765
– Question Attention	0.6847	0.6845
– Answer Attention	0.6693	0.6703
– Conditional Attention	0.6815	0.6795
– All Attention	0.6555	0.6588

Table 3: An ablation test of RCA model on the test set. Macro and micro average F1 scores are presented.

portance weights of question words. As shown in Figure 3, for the question, AoA assigns the largest weight to the word “has” while RCA emphasizes on “benefits”. For the answer, RCA assigns the largest weight to key word “promote” and only RCA correctly predicts the right stance *favor*.

### Influence of Maximum Reading Steps

Figure 4 describes the performance of RCA with respect to maximum reading steps. Among the displayed models, we can find that reading the question-answer pairs for multiple times is always better than only once. By conducting multiple conditional attention steps, RCA is able to supplement the missing clues for final stance prediction. When reading the context for 9 times, the RCA achieves best performance both on *macro-* and *micro-* *F1* scores. Similar to previous models based on Memory Networks with “multi-hop attention” (Sukhbaatar et al. 2015; Tang, Qin, and Liu 2016; Chen et al. 2017), adding the number of reading steps doesn’t monotonically contribute to the performance but increase running time as well as the risk of over-fitting.

### Ablation Test of RCA

To further explore how well each part contributes to predicting stance, we conduct an ablation test of the RCA model on test set. From Table 3, we can tell that modeling question words with GRU instead of word embeddings in RCA contributes about 2.71% *macro F1* score and 2.72% *micro F1* score, showing the necessity of better semantic composition of questions. Since AnswerStance is a target-dependent

stance classification problem, attending answer with question information is more important than attending question with answer information.

### Discussion of Conditional Attention

While previous memory networks (Sukhbaatar et al. 2015; Tang, Qin, and Liu 2016; Chen et al. 2017) usually first encode a target or question into a fixed-sized vector and keep it unchanged during multiple attention hops. In RCA, the recurrent reading steps is deeply coupled with conditional attention structure, where the question representations are polished regarding to the previous stance states which contain mutual relationships among question, answer and stance. This makes two consequent reading steps more similar to the idea of *attention-over-attention*.

However, different from AoA model (Cui et al. 2017), the conditional attention structure in RCA utilizes a stance vector as the proxy to attend the question and answer, where answer words are first conditioned on the stance states with most recently updated target representation and then the question words are conditioned on next stance state with latest added answer information. Thereby, the stance vector can serve as a good semantic summary of the question-answer pair, which is proven to be more suitable than other mingling methods for the answer stance detection task. Besides, through recurrent iterations, RCA obtains the ability of deliberating stance orientation by gradually abstracting in-depth semantic representations of question-answer pairs, which significantly contributes to making more accurate predictions of stance labels.

### Case Study

We visualize the prediction process of RCA with 3 reading steps in Figure 5. For the left example, we can see that RCA gradually captures important words like “German” from question and “not likely” from answer. For the right example, RCA places higher importance to “near” which implies the existence of the hospital step by step. These two examples demonstrates the effectiveness of our model.

### Related Work

In this section, we briefly review related work on stance detection task and target dependent models.

### Stance Detection

Previous works mostly focus on identifying stance in debates (Hasan and Ng 2013; Sridhar et al. 2015), news articles (Ferreira and Vlachos 2016), tweets (Mohammad et al. 2016), micro-blogs (Xu et al. 2016), fake news (Bhatt et al. 2018) where one classifies the stance of the body text relative to the claim made in the headline, and rumour (Qazvinian et al. 2011; Derczynski et al. 2017) where systems should determine how each post is oriented with regard to the rumor’s veracity. Our task distinguishes from those stance classification task in two ways: 1) we detect stance embedded in certain types of question-answer pairs. 2) the targets here are not given entities or claims but entire questions and therefore are harder to depict.

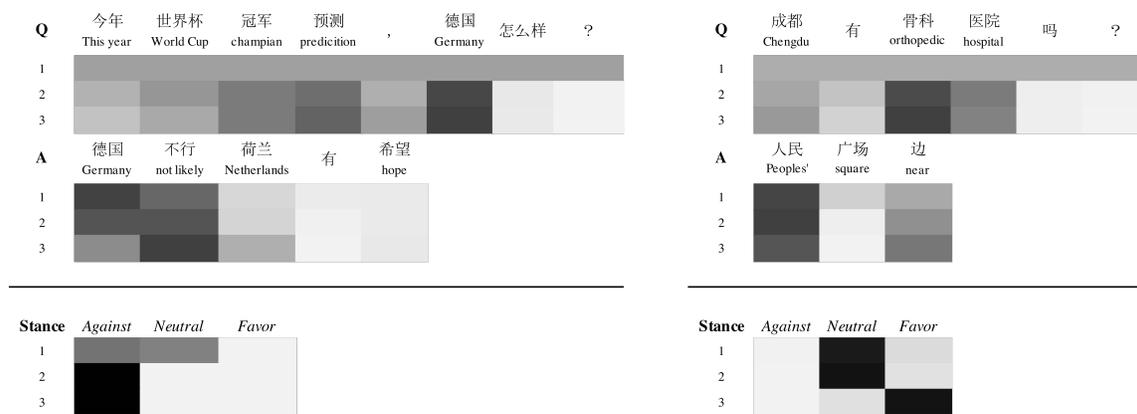


Figure 5: Visualization of stance state at each reading step through a softmax classifier and attention weights of each question and answer word for the examples. Deeper color implies larger probability. A RCA model with 3 reading steps is used for illustration.

Similarly, we also tackle stance detection in community QA as a generalized target dependent stance detection task, in which the question can be seen as the target and answer as the context.

### Target-Dependent Models

Various deep learning approaches have been proposed to solve the target dependent sentiment classification and stance detection problem. Tang *et al.*(2016) used target-dependent LSTM (TD-LSTM) to model the preceding and following contexts surrounding the target so as to learn target dependent representations of review sentences. Wang *et al.*(2016) and Du *et al.*(2017) incorporate attention mechanism into LSTM to acquire target dependent review representation in sentiment analysis and tweet representations in stance detection respectively. Our model is partly motivated by (Tang, Qin, and Liu 2016; Chen et al. 2017) which employ memory networks with multi-hop attention to learn target relevant sentence presentation for sentiment analysis. Though achieving excellent performance on previous target dependent tasks, most of these memory networks only encode the targets once without modifying these representations with information from review or answer text. Hence, they are not capable of capturing the complex interaction between the target question and answer sentence in Answer-Stance.

### Conclusion

In this paper, we develop the RCA model for answer stance detection in community QA. The RCA model can effectively represent questions and answers with mutual conditional attention and can polish the stance states through recurrent reading. Experiments on a stance-annotated Chinese Community QA dataset demonstrate that RCA model outperforms other competitive baseline methods.

For future work, we would like to investigate automatic control of maximum reading steps based on the complexity of given QA pairs with reinforcement learning techniques.

### Acknowledgments

We would like to thank all the anonymous reviewers for their insightful comments. Meanwhile, we thank Dr. Xi-aocheng Feng, Prof. Wanxiang Che and Dr. Weinan Zhang for their help on revising the manuscript. The paper is supported by National Natural Science Foundation of China (No. 61632011, No. 61772153 and No. 71490722). The paper is also supported by Sogou Technology Inc.

### References

- Augenstein, I.; Rocktäschel, T.; Vlachos, A.; and Bontcheva, K. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 876–885.
- Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *ArXiv e-prints*.
- Bhatt, G.; Sharma, A.; Sharma, S.; Nagpal, A.; Raman, B.; and Mittal, A. 2018. Combining neural, statistical and external features for fake news stance identification. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, 1353–1357. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.
- Che, W.; Li, Z.; and Liu, T. 2010. Ltp: A chinese language technology platform. In *Coling 2010: Demonstrations*, 13–16.
- Chen, P.; Sun, Z.; Bing, L.; and Yang, W. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 452–461.
- Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1724–1734.

- Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; Liu, T.; and Hu, G. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 593–602. Vancouver, Canada: Association for Computational Linguistics.
- Derczynski, L.; Bontcheva, K.; Liakata, M.; Procter, R.; Wong Sak Hoi, G.; and Zubiaga, A. 2017. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 69–76. Vancouver, Canada: Association for Computational Linguistics.
- Du, J.; Xu, R.; He, Y.; and Gui, L. 2017. Stance classification with target-specific neural attention. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 3988–3994.
- Elman, J. L. 1990. Finding structure in time. *Cognitive Science* 14(2):179–211.
- Ferreira, W., and Vlachos, A. 2016. Emergent: a novel dataset for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1163–1168.
- Hasan, K. S., and Ng, V. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 1348–1356.
- Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.
- Mohammad, S.; Kiritchenko, S.; Sobhani, P.; Zhu, X.; and Cherry, C. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, 31–41.
- Nakov, P.; Màrquez, L.; Magdy, W.; Moschitti, A.; Glass, J.; and Randeree, B. 2015. Semeval-2015 task 3: Answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 269–281. Denver, Colorado: Association for Computational Linguistics.
- Qazvinian, V.; Rosengren, E.; Radev, D. R.; and Mei, Q. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1589–1599.
- Sobhani, P.; Mohammad, S.; and Kiritchenko, S. 2016. Detecting stance in tweets and analyzing its interaction with sentiment. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, 159–169.
- Sridhar, D.; Foulds, J.; Huang, B.; Getoor, L.; and Walker, M. 2015. Joint models of disagreement and stance in online debate. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 116–125.
- Sukhbaatar, S.; szlam, a.; Weston, J.; and Fergus, R. 2015. End-to-end memory networks. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc. 2440–2448.
- Tang, D.; Qin, B.; Feng, X.; and Liu, T. 2016. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING 2016*, 3298–3307.
- Tang, D.; Qin, B.; and Liu, T. 2016. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 214–224.
- Vijayaraghavan, P.; Sysoev, I.; Vosoughi, S.; and Roy, D. 2016. Deepstance at semeval-2016 task 6: Detecting stance in tweets using character and word-level cnns. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 413–419. San Diego, California: Association for Computational Linguistics.
- Wang, Y.; Huang, M.; zhu, x.; and Zhao, L. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 606–615.
- Wei, W.; Zhang, X.; Liu, X.; Chen, W.; and Wang, T. 2016. pkudblab at semeval-2016 task 6 : A specific convolutional neural network system for effective stance detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, 384–388.
- Xu, R.; Zhou, Y.; Wu, D.; Gui, L.; Du, J.; and Xue, Y. 2016. Overview of nlpcc shared task 4: Stance detection in chinese microblogs. In Lin, C.-Y.; Xue, N.; Zhao, D.; Huang, X.; and Feng, Y., eds., *Natural Language Understanding and Intelligent Applications*, 907–916. Cham: Springer International Publishing.