Aspect Sentiment Classification Towards Question-Answering with Reinforced Bidirectional Attention Network

Jingjing Wang[1], **Changlong Sun**[2], **Shoushan Li**[1,*], **Xiaozhong Liu**[2],
Min Zhang[1], **Luo Si**[2], **Guodong Zhou**[1]
[1]School of Computer Science and Technology, Soochow University, China
[2]Alibaba Group, China
djingwang@gmail.com, {lishoushan, minzhang, gdzhou}@suda.edu.cn,
{changlong.scl, xiaozhong.lxz, luo.si}@alibaba-inc.com

Abstract

In the literature, existing studies on aspect sentiment classification (ASC) focus on individual non-interactive reviews. This paper extends the research to interactive reviews and proposes a new research task, namely Aspect Sentiment Classification towards Question-Answering (ASC-QA), for real-world applications. This new task aims to predict sentiment polarities for specific aspects from interactive QA style reviews. In particular, a high-quality annotated corpus is constructed for ASC-QA to facilitate corresponding research. On this basis, a Reinforced Bidirectional Attention Network (RBAN) approach is proposed to address two inherent challenges in ASC-QA, i.e., semantic matching between question and answer, and data noise. Experimental results demonstrate the great advantage of the proposed approach to ASC-QA against several state-of-the-art baselines.

1 Introduction

As a fine-grained sentiment analysis task, Aspect Sentiment Classification (ASC) aims to predict sentiment polarities (e.g., *positive*, *negative*, *neutral*) towards given particular aspects from a text and has been drawing more and more interests in natural language processing and computational linguistics over the past few years (Jiang et al., 2011; Tang et al., 2016b; Wang et al., 2018a). However, most of the existing studies on ASC focus on individual non-interactive reviews, such as customer reviews (Pontiki et al., 2014) and tweets (Mitchell et al., 2013; Vo and Zhang, 2015; Dong et al., 2014). For example, in a customer review "*The food is delicious, but ambience is badly in need of improvement.*", the customer mentions two aspects, i.e., "*food*" and "*ambience*", and expresses *positive* sentiment towards the former and *negative* sentiment towards the latter.

Question-Answering (QA) Style Review

- Question: Is **[battery life]** durable? How about **[operating speed]** of the phone?
- Answer: Yes, very durable but quite slow and obtuse.

Aspect Sentiment Classification Towards QA

- Input: QA text pair with given aspects
- Output: **[battery life]:** *Positive*
[operating speed]: *Negative*

Figure 1: An example for illustrating the proposed task of Aspect Sentiment Classification towards Question-Answering (ASC-QA).

Recently, a new interactive reviewing form, namely "*Customer Question-Answering (QA)*", has become increasingly popular and a large-scale of such QA style reviews (as shown in Figure 1) could be found in several famous e-commerce platforms (e.g., *Amazon* and *Taobao*). Compared to traditional non-interactive customer reviews, interactive QA style reviews are more reliable and convincing because answer providers are randomly selected from the real customers who have purchased the product (Shen et al., 2018a). To well automatically-understand the QA style reviews, it's worthwhile to perform ASC on the QA style reviews.

However, we believe that Aspect Sentiment Classification towards QA (ASC-QA) is not easy work and this novel task faces at least two major challenges. On one hand, different from traditional non-interactive reviews with a single sequence structure, interactive QA style reviews consist of two parallel units, i.e., question and answer. Thus, it's rather difficult to infer the sentiment polarity towards an aspect based on a single question or single answer. Take Figure 1 as an example. A well-behaved approach to ASC-QA should match each question and answer bidirectionally so as to correctly determine the sentiment polarity towards a specific aspect.

[*] Corresponding author

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3548–3557
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

On the other hand, different from common QA matching tasks such as question-answering (Shen et al., 2018a), ASC-QA focuses on extracting sentiment information towards a specific aspect and may suffer from much aspect-irrelevant noisy information. For instance, in Figure 1, although the words in the answer (e.g., "*quite slow*", "*obtuse*") and the question (e.g., "*operating speed*") are relevant to aspect "*operating speed*", they are noisy for the other aspect "*battery life*". These noisy words might provide wrong signals and mislead the model into assigning a *negative* sentiment polarity to aspect "*battery life*" and vice versa. Therefore, a well-behaved approach to ASC-QA should alleviate the effects of noisy words for a specific aspect in both question and answer during model training.

In this paper, we propose a reinforced bidirectional attention network approach to tackle the above two challenges. Specifically, we first propose a word selection model, namely Reinforced Aspect-relevant Word Selector (RAWS), to alleviate the effects of noisy words for a specific aspect through discarding noisy words and only select aspect-relevant words in a word sequence. On the basis of RAWS, we then develop a Reinforced Bidirectional Attention Network (RBAN) approach to ASC-QA, which employs two fundamental RAWS modules to perform word selection over the question and answer text respectively. In this way, RBAN is capable of not only addressing the semantic matching problem in the QA text pair, but also alleviating the effects of noisy words for a specific aspect in both the question and answer sides. Finally, we optimize RBAN via a reinforcement learning algorithm, i.e., policy gradient (Williams, 1992; Sutton et al., 1999). The main contributions of this paper are in two folds:

- We propose a new research task, i.e., Aspect Sentiment Classification towards Question-Answering (ASC-QA), and construct a high-quality annotated benchmark corpus for this task.
- We propose an innovative reinforced bidirectional attention network approach to ASC-QA and validate the effectiveness of this approach through extensive experiments.

2 Data Collection and Annotation

We collect 150k QA style reviews from *Taobao*[1], the most famous electronic business platform in China. The QA style reviews consist of three different domains: *Bags*, *Cosmetics* and *Electronics*. Since corpus annotation is labor-expensive and time-consuming, we randomly select 10k QA text pairs from each domain to perform annotation. Specifically, following Pontiki et al. (2014), we define an aspect at two levels of granularity, i.e., aspect term and aspect category. Besides, following Pontiki et al. (2015), we define three sentiment polarities, i.e., *positive*, *negative* and *neutral* (mildly positive or mildly negative) towards both aspect terms and categories. In this way, each QA text pair is annotated with two tuples, i.e., (*aspect term*, *polarity*), (*aspect category*, *polarity*).

For Tuple (*Aspect Term*, *Polarity*), we annotate the single/multi-word terms together with its corresponding polarities inside each QA text pair according to four main guidelines as follows:

(1) We only annotate the aspect term when the related question and answer are matched. For example, the QA text pair in Figure 1 is annotated as ("*battery life*", *positive*) and ("*operating speed*", *negative*) due to words "*durable*", "*slow*" and "*obtuse*". However, in **E1**, the answer does not reply to the question correctly and thus the aspects of "*macos*" and "*screen*" will not be annotated.

E1: Q: *Is macos good? How about the screen?*
A: *The shopkeeper is very warm-hearted.*

(2) We only annotate the aspect term towards which an opinion is expressed. For example, in **E2**, the answer conveys only objective information without expressing opinions towards "*phone*" and thus "*phone*" will not be annotated. However, "*case*" will be annotated and tagged as *neutral*.

E2: Q: *How is this phone? How about the case?*
A: *I bought this phone yesterday. Case is okay nothing great.*

(3) We only annotate aspect terms which explicitly name particular aspects. For example, in **E3**, "*this*", "*it*" will not be annotated.

E3: Q: *Is this expensive? Did anybody buy one?*
A: *Of course, it's quite expensive.*

(4) When one aspect term has two different descriptions in both question and answer, the annotated aspect term should be consistent with the question. For example, in **E4**, the annotated aspect term should be "*battery life*" instead of "*battery*".

E4: Q: *Is battery life durable?*
A: *Yes, this battery is very durable.*

[1] http://www.taobao.com

Domains	Aspect Categories
Bags	*Size*, *Price*, *Appearance*, *Quality*, *Weight*, *Certified Products*, *Smell*, *Accessories*, *Material*, *Life Timer*, *Style*, *Workmanship*, *Color*, *Stain Resistant*, *Practicality*
Cosmetics	*Price*, *Efficacy*, *Moisturizing Performance*, *Certified Products*, *Adverse Reaction*, *Exfoliator*, *Texture*, *Long Lasting*, *Smell*, *Material*, *Noticeable Color*, *Quality*, *Colour*, *Touch*, *Skin Whitening*, *Acne*
Electronics	*System Performance*, *Appearance*, *Battery*, *Computing (e.g., cpu, gpu, tpu etc.)*, *Certified Products*, *Quality*, *IO (e.g., keyboard, screen, etc.)*, *Price*, *Storage*, *Function (e.g., touch id, waterproof etc.)*

Table 1: The defined aspect categories in each domain.

Domains	*Pos.*	*Neg.*	*Neu.*	*All*	*#Cat.*
Bags	2503	724	453	3680	15
Cosmetics	2834	956	503	4293	16
Electronics	2742	821	531	4094	10

Table 2: Corpus statistics (*Pos.*, *Neg.* and *Neu.* denote the number of *positive*, *negative* and *neutral* for aspect term; *#Cat.* denotes the number of aspect category).

For Tuple (*Aspect Category*, *Polarity*), we first define[2] 15, 16, 10 aspect categories (as shown in Table 1) for the domains of *Bags*, *Cosmetics* and *Electronics* respectively. Then, we annotate aspect categories (chosen from the above predefined category list) discussed in each QA text pair according to similar guidelines for aspect term. For example, there are two aspect categories discussed in Figure 1, i.e., *Battery* and *System Performance*, and annotated as (*Battery*, *positive*) and (*System Performance*, *negative*) respectively. Finally, we discard the QA text pairs which have no annotated term and category.

We assign two annotators to tag each QA text pair and the *Kappa* consistency check value of the annotation is 0.81. When two annotators cannot reach an agreement, an expert will make the final decision, ensuring the quality of data annotation. Table 2 shows the statistics of the final corpus. To motivate future investigations for this track of research, the annotated corpus consisting of three domains are released in github[3].

3 Our Approach

In this section, we first introduce the word selection model, i.e., Reinforced Aspect-relevant Word Selector (RAWS) as illustrated in Figure 2, which functions as a fundamental module of our approach to alleviate the effects of noisy words (Section 3.1). On the basis of RAWS, we present the Reinforced Bidirectional Attention Network (RBAN) approach to ASC-QA as illustrated in Figure 3, which employs two RAWS modules to perform word selection over the question and answer text respectively (Section 3.2). Finally, we introduce our optimization strategy via policy gradient and back-propagation (Section 3.3).

3.1 Reinforced Aspect-relevant Word Selector (RAWS)

Figure 2 shows the framework of the word selection model, i.e., Reinforced Aspect-relevant Word Selector (RAWS). Given an **input** word sequence $x = \{x_1, .., x_E\}$, RAWS aims to discard noisy words and only select aspect-relevant words inside x for a specific aspect $\boldsymbol{x_{aspect}}$[4]. The **output** of RAWS is an equal-length sequence of one-hot variables $o = [o_1, .., o_E]$, where $o_i = 1$ if the word x_i is selected otherwise $o_i = 0$.

In this way, RAWS virtually functions as a "hard" attention mechanism and thus cannot be directly optimized through back-propagation due to the non-differentiable problem as proposed in Xu et al. (2015) and Shen et al. (2018b). To address this issue, we employ the reinforcement learning algorithm, i.e., policy gradient (Sutton et al., 1999), to model RAWS. In this fashion, RAWS plays as an agent which decides to select the word or not by following a policy network as follows.

Policy Network. In this paper, we adopt a stochastic policy network p_π which can provide a conditional probability distribution $p_\pi(o|\cdot)$ over action sequence $o = [o_1, .., o_E]$. Here, o is exactly the output of RAWS and $o_i = 1$ indicates that x_i is selected otherwise $o_i = 0$ indicates that x_i is discarded. More specifically, we adopt LSTM (Graves, 2013) to construct the policy network p_π for performing word selection over word sequence x, denoted as LSTM_p. In order to differentiate whether a word is selected or discarded, inspired by Lei et al. (2016), we incorporate the action result o_i into the input $\hat{v}_i$ of LSTM_p at time-step i and compute hidden state $h_i \in \mathbb{R}^d$ of word x_i as:

$$h_i = \text{LSTM}_p(\hat{v}_i), \;\; \hat{v}_i = v_i \oplus (o_i \otimes e) \qquad (1)$$

[2] Aspect categories are defined and summarized through preliminary annotation.

[3] https://github.com/jjwangnlp/ASC-QA

[4] The aspect denotes an *aspect term* or *aspect category* as introduced in Section 2.

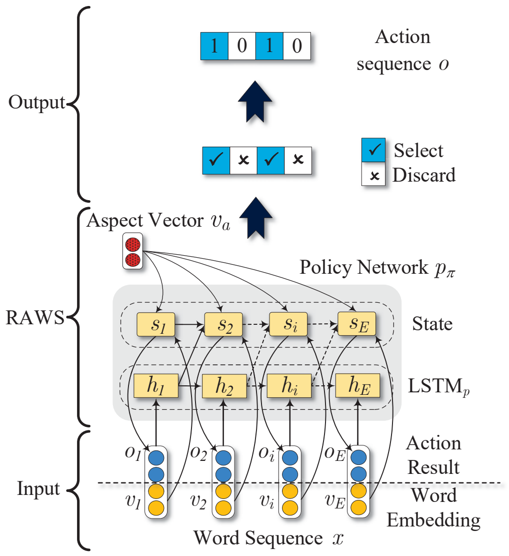


Figure 2: The framework of word selection model, i.e., Reinforced Aspect-relevant Word Selector (**RAWS**).

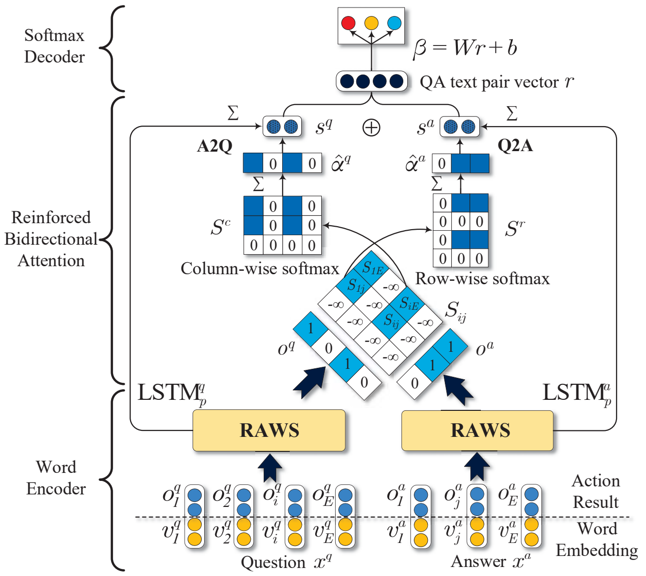


Figure 3: The framework of our proposed Reinforced Bidirectional Attention Network (**RBAN**) approach.

where $v_i \in \mathbb{R}^d$ is word embedding of word x_i; $\oplus$ denotes vector concatenation and $\otimes$ denotes element-wise multiplication; $o_i \otimes e = [o_i; ..; o_i]$, that is, o_i is tiled d' times across the row, where $e \in \mathbb{R}^{d'}$ is a column vector with d' 1s and d' is set to be 50 tuned with development set; $\hat{v}_i \in \mathbb{R}^{d+d'}$.

In principle, the policy network p_π uses a **Reward** to guide the policy learning over word sequence x. It samples an **Action** o_i with the probability $p_\pi(o_i|s_i;\theta_r)$ at each **State** s_i. In this paper, state, action and reward are defined as follows.

- **State.** The state s_i at i-th time-step should provide adequate information for deciding to select a word or not for aspect $\boldsymbol{x_{aspect}}$. Thus, the state $s_i \in \mathbb{R}^{4d}$ is composed of four parts, i.e., h_{i-1}, c_{i-1}, v_i and v_a, defined as $s_i = h_{i-1} \oplus c_{i-1} \oplus v_i \oplus v_a$, where c_{i-1} is memory state of LSTM$_p$; $v_a \in \mathbb{R}^d$ is aspect vector[5] of $\boldsymbol{x_{aspect}}$.
- **Action.** p_π samples *action* $o_i \in \{0, 1\}$ with conditional probability $p_\pi(o_i|s_i;\theta_r)$, which could be cast as a binary classification problem. Thus, we use a logistic function to compute $p_\pi(o_i|s_i;\theta_r)$.

$$\begin{aligned} o_i \sim p_\pi(o_i|s_i;\theta_r) &= o_i \, \mathrm{sigmoid}(W_r s_i + b_r) \\ &+ (1 - o_i)(1 - \mathrm{sigmoid}(W_r s_i + b_r)) \end{aligned} \quad (2)$$

where $\theta_r = \{W_r \in \mathbb{R}^{1\times 4d}, b_r \in \mathbb{R}\}$ is the parameter to be learned. $\sim$ denotes the discrete *action* sampling operation.

- **Reward.** In order to select aspect-relevant words inside word sequence x, we define an aspect-relevant reward $\mathcal{R}$ based on *cosine* similarity between aspect vector $v_a \in \mathbb{R}^d$ of $\boldsymbol{x_{aspect}}$ and the last hidden state $h_E \in \mathbb{R}^d$ of LSTM$_p$ after p_π finishes all actions, i.e.,

$$\begin{aligned} \mathcal{R} &= \log\cos(v_a, h_E) \\ &+ \log p(y|(\mathcal{P}, \boldsymbol{x_{aspect}})) - \gamma E'/E \end{aligned} \quad (3)$$

where $\log\cos(v_a, h_E) = \log \frac{v_a \cdot h_E}{||v_a||\,||h_E||}$ is a *cosine* delay reward. Besides, it's worthwhile to mention that, we regard the loss $\log p(y|(\mathcal{P}, \boldsymbol{x_{aspect}})$ presented in Eq.(10) from the classification phase as another loss delay reward. This loss reward combining with the above *cosine* reward could provide adequate supervision signals to guide RAWS to select aspect-relevant and also discriminative words (e.g., sentiment words "*slow*" and "*obtuse*" for aspect "*operating speed*") for performing ASC-QA. $\gamma E'/E$ is an additional term for limiting the number of selected words. $E' = \sum_{i=1}^{E} o_i$ denotes the number of selected words. γ is a penalty weight (tuned to be 0.01 with development set).

3.2 Reinforced Bidirectional Attention Network (RBAN)

Figure 3 shows the overall framework of our proposed reinforced bidirectional attention network (RBAN) approach to ASC-QA, which consists of three parts: **1)** Word Encoder. **2)** Reinforced Bidirectional Attention. **3)** Softmax Decoder.

Word Encoder. Given a QA text pair $\mathcal{P}$ with an aspect $\boldsymbol{x_{aspect}}$, let $x^q = \{x_i^q\}, \forall i \in [1, E^q]$ denotes the word sequence in question text, and $x^a = \{x_j^a\}, \forall j \in [1, E^a]$ denotes the word sequence in answer text. To alleviate the effects of

[5] If aspect is a single word like *"food"*, aspect vector is word embedding, while aspect is multi-word expression like *"operating speed"* in Figure 1, aspect vector is an average of its constituting word embeddings as Tang et al. (2016b).

noisy words for aspect $\boldsymbol{x_{aspect}}$ in both the question and answer text, we make use of two RAWS modules (as introduced in Section 3.1) to perform word selection over question x^q and answer x^a respectively. More specifically, we employ two LSTM_p to construct policy networks p^q_π and p^a_π for sampling action o^q over question x^q and sampling action o^a over answer x^a. Here, the two LSTM_p are denoted as LSTM^q_p and LSTM^a_p respectively. Therefore, according to Eq.(1), the hidden states $h^q_i, h^a_j \in \mathbb{R}^d$ of words x^q_i and x^a_j are computed as:

$$\begin{aligned} h^q_i &= \text{LSTM}^q_p(\hat{v}^q_i), \ \ \hat{v}^q_i = v^q_i \oplus (o^q_i \otimes e) \\ h^a_j &= \text{LSTM}^a_p(\hat{v}^a_j), \ \ \hat{v}^a_j = v^a_j \oplus (o^a_j \otimes e) \end{aligned} \tag{4}$$

where $v^q_i, v^a_j \in \mathbb{R}^d$ are word embeddings (presented in Section 4.1) of the word x^q_i and x^a_j.

Reinforced Bidirectional Attention. Once two RAWS modules finish all their actions $o^q = [.., o^q_i, ..]$ and $o^a = [.., o^a_j, ..]$ over question x^q and answer x^a, we employ a positional mask matrix $M \in \mathbb{R}^{E^q \times E^a}$ to calculate the matching matrix $S \in \mathbb{R}^{E^q \times E^a}$ between question and answer as:

$$M_{ij} = \begin{cases} 0 & o^q_i = o^a_j = 1 \\ -\infty & \text{otherwise} \end{cases} \tag{5}$$

$$S_{ij} = w^\top \tanh(W_1 h^q_i + W_2 h^a_j + b) + M_{ij} \tag{6}$$

where S_{ij} denotes the similarity between the i-th question word and the j-th answer word; $M_{ij} = -\infty$ leads to $S_{ij} = -\infty$ indicating that the i-th question word or the j-th answer word has been regarded as the noisy word for $\boldsymbol{x_{aspect}}$ and thus discarded by RAWS; $W_1, W_2 \in \mathbb{R}^{d \times d}, w, b \in \mathbb{R}^d$ are the trainable parameters.

In order to mine semantic matching information between question and answer, we employ S to compute attentions in both directions, which could be seen as a Question-to-Answer attention and an Answer-to-Question attention. Specifically, we first employ the row/column-wise softmax operation to get two normalized matrices S^r and S^c.

$$\begin{aligned} S^r_{i:} &= \text{softmax}([S_{i1}, .., S_{iE^a}]), \ \forall i \in [1, E^q] \\ S^c_{:j} &= \text{softmax}([S_{1j}, .., S_{E^q j}]), \ \forall j \in [1, E^a] \end{aligned} \tag{7}$$

where $S_{ij} = -\infty$ leads to ${S^r}_{ij}, {S^c}_{ij} = 0$ when the softmax operation is applied. This switches off the attentions between word x^q_i and x^a_j so as to filter the noisy word information and only mine the matching information relevant to aspect $\boldsymbol{x_{aspect}}$.

Second, since each word x^q_i in question interacts all words in answer x^a and vice versa, its importance can be measured as the summation of the strengths of all these interactions, i.e., matching scores computed in Eq.(7). Therefore, we perform row/column-wise summation operation over the normalized matching matrices, i.e., $\hat{\alpha}^a = \sum_i S^r_{i:}$ and $\hat{\alpha}^q = \sum_j S^c_{:j}$, where $\hat{\alpha}^a = [.., \hat{\alpha}^a_j, ..] \in \mathbb{R}^{E^a}$ and $\hat{\alpha}^q = [.., \hat{\alpha}^q_i, ..] \in \mathbb{R}^{E^q}$ are matching score vectors. Finally, the bidirectional attention is computed as follows:

- **Question-to-Answer Attention (Q2A).** We first perform softmax operation over $\hat{\alpha}^a$ to compute the attention weight α^a_j of word x^a_j in answer text as $\alpha^a_j = \frac{\exp(\hat{\alpha}^a_j)}{\sum_{t=1}^{E^a} \exp(\hat{\alpha}^a_t)}$. Then, the vector $s^a \in \mathbb{R}^d$ of the answer text is computed as a weighted sum of hidden state h^a_j based on the attention weight α^a_j, i.e., $s^a = \sum_{j=1}^{E^a} \alpha^a_j h^a_j$.
- **Answer-to-Question Attention (A2Q).** Similar to question-to-answer attention, the question vector $s^q \in \mathbb{R}^d$ is computed based on attention weight $\alpha^q_i = \frac{\exp(\hat{\alpha}^q_i)}{\sum_{t=1}^{E^q} \exp(\hat{\alpha}^q_t)}$, i.e., $s^q = \sum_{i=1}^{E^q} \alpha^q_i h^q_i$.

Subsequently, we concatenate the answer vector s^a and question vector s^q so as to obtain the vector representation $r \in \mathbb{R}^{2d}$ of the QA text pair $\mathcal{P}$, i.e., $r = s^a \oplus s^q$.

Softmax Decoder. To perform ASC-QA, we feed the vector r to a softmax classifier, i.e., $\beta = Wr + b$, where $\beta \in \mathbb{R}^C$ is the output vector. Then, the probability of labeling sentence with sentiment polarity $l \in [1, C]$ is computed by $p_\theta = \frac{\exp(\beta_l)}{\sum_{c=1}^{C} \exp(\beta_c)}$. Finally, the label with the highest probability stands for the predicted sentiment polarity towards aspect $\boldsymbol{x_{aspect}}$.

3.3 Optimization via Policy Gradient and Back-Propagation

The parameters in RBAN are divided into two groups: **1)** θ^q_r and θ^a_r for policy networks p^q_π, p^a_π in two fundamental RAWS modules. **2)** θ for the rest parts including word embeddings, LSTM, bidirectional attention and softmax decoder.

For θ^q_r, we optimize it with policy gradient algorithm (Sutton et al., 1999). In detail, we first obtain an aspect-relevant reward $\mathcal{R}^q$ according to Eq.(3) after p^q_π finishes all actions. Then, the policy gradient w.r.t. θ^q_r is computed by differentiating the maximized expected reward $J(\theta^q_r)$ as follows:

$$\nabla_{\theta^q_r} J(\theta^q_r) = \mathbb{E}_{o^q \sim p^q_\pi}\Big[\sum_{i=1}^{E^q} \mathcal{R}^q \nabla_{\theta^q_r} \log p^q_\pi(o^q_i | s^q_i)\Big] \tag{8}$$

where $\nabla_{\theta_r^q} J(\theta_r^q)$ is estimated by using Monte-Carlo simulation (Sutton et al., 1999) to sample some *action* sequences over question texts. Similarly, the policy gradient w.r.t. θ_r^a is computed as:

$$\nabla_{\theta_r^a} J(\theta_r^a) = \mathbb{E}_{o^a \sim p_\pi^a}[\sum_{j=1}^{E^a} \mathcal{R}^a \nabla_{\theta_r^a} \log p_\pi^a(o_j^a|s_j^q)] \tag{9}$$

For θ, we optimize it with back-propagation. In detail, the objective of learning θ is to minimize the cross-entropy loss function in the classification phase as follows:

$$J(\theta) = \mathbb{E}_{(\mathcal{P}, \boldsymbol{x_{aspect}}, y) \sim \mathcal{D}}[-\log p(y|(\mathcal{P}, \boldsymbol{x_{aspect}}))] \tag{10}$$

where $(\mathcal{P}, \boldsymbol{x_{aspect}}, y)$ denotes QA text pair $\mathcal{P}$ with given aspect $\boldsymbol{x_{aspect}}$ from dataset $\mathcal{D}$; y is ground-truth sentiment polarity towards aspect $\boldsymbol{x_{aspect}}$.

Note that, during model training, θ_r^q and θ_r^q are not updated in early stage, and thus two RAWS modules select all words in question and answer. When θ is optimized until the loss over development set does not decrease significantly, we then begin to optimize θ, θ_r^q and θ_r^a simultaneously.

4 Experimentation

We systematically evaluate the performance of our proposed RBAN approach to ASC-QA on the corpus as described in Section 2.

4.1 Experimental Settings

Data Settings. As introduced in Section 2, we have annotated QA text pairs from three different domains listed in Table 2. For each domain, we randomly split the annotated data into training, development, and testing sets with the ratio of 8:1:1.

Word Embedding. We first adopt FudanNLP (Qiu et al., 2013) to perform word segmentation over our collected 150k Chinese QA text pairs. Then, we employ these QA text pairs to pre-train 200-dimension word vectors with skip-gram[6].

Hyper-parameters. In all our experiments, word embeddings are optimized during training. The dimensions of LSTM hidden states are set to be 200. The other hyper-parameters are tuned according to the development set. Specifically, we adopt Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.01 for cross-entropy training and adopt the SGD optimizer with a learning rate of 0.002 for all policy gradients training. Regularization weight of parameters is 10^{-5}, dropout rate is 0.25 and batch size is 32.

Evaluation Metrics. The performance is evaluated using *Accuracy* (Acc.) and *Macro-F1* (F1) (Wang et al., 2018a). Moreover, t-test is used to evaluate the significance (Yang and Liu, 1999).

Task Definition. Our proposed ASC-QA consists of two sub-tasks: **1) Term-level ASC-QA.** Given a set of pre-identified aspect terms, this sub-task is to determine the polarity towards each aspect term inside a QA text pair. **2) Category-level ASC-QA.** Given a set of pre-identified aspect categories, this sub-task is to determine the polarity towards each aspect category discussed in a QA text pair.

4.2 Baselines

For comparison, we implement several state-of-the-art approaches to ASC as baselines. Since the input of all these approaches should be a single sequence, we concatenate question and answer text to generate a single sequence. Besides, we employ some QA matching approaches to ASC-QA and implement several basic versions of RBAN as baselines. Note that, for fair comparison, all the above baselines adopt the same pre-trained word embeddings as RBAN.

The baselines are listed as follows in detail: **1) LSTM** (Wang et al., 2016). This approach only adopts a standard LSTM network to model the text without considering aspect information. **2) RAM** (Chen et al., 2017). This is a state-of-the-art deep memory network approach to ASC. **3) GCAE** (Xue and Li, 2018). This is a state-of-the-art approach to ASC which combines CNN and gating mechanisms to learn text representation. **4) S-LSTM** (Wang and Lu, 2018). This is a state-of-the-art approach to ASC which considers structural dependencies between targets and opinion terms. **5) BIDAF** (Seo et al., 2016). This is a QA matching approach to reading comprehension. We substitute its decoding layer with softmax decoder to perform ASC-QA. **6) HMN** (Shen et al., 2018a). This is a QA matching approach to coarse-grained sentiment classification towards QA style reviews. **7) MAMC** (Yin et al., 2017). This is a QA matching approach to ASC which proposes a hierarchical iterative attention to learn the aspect-specific text representation. **8) RBAN w/o RAWS**. Our RBAN approach without using RAWS modules. **9) RBAN w/o Q2A**. Our RBAN

[6] https://github.com/dav/word2vec

Approaches	**Term-level ASC-QA**						**Category-level ASC-QA**					
	Bags		*Cosmetics*		*Electronics*		*Bags*		*Cosmetics*		*Electronics*	
	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.
LSTM (Wang et al., 2016)	0.571	0.757	0.582	0.771	0.534	0.756	0.528	0.773	0.493	0.739	0.522	0.752
RAM (Chen et al., 2017)	0.605	0.782	0.614	0.805	0.557	0.788	0.561	0.795	0.519	0.762	0.579	0.792
GCAE (Xue and Li, 2018)	0.617	0.779	0.623	0.819	0.570	0.781	0.590	0.787	0.514	0.791	0.576	0.788
S-LSTM (Wang and Lu, 2018)	0.615	0.824	0.623	0.821	0.569	0.794	0.587	0.828	0.522	0.788	0.581	0.801
BIDAF (Seo et al., 2016)	0.613	0.815	0.618	0.813	0.558	0.809	0.592	0.830	0.515	0.788	0.571	0.787
HMN (Shen et al., 2018a)	0.607	0.817	0.615	0.821	0.561	0.802	0.606	0.827	0.512	0.798	0.579	0.804
MAMC (Yin et al., 2017)	0.621	0.825	0.629	0.823	0.562	0.815	0.612	0.837	0.524	0.794	0.582	0.805
RBAN w/o RAWS	0.623	0.826	0.633	0.827	0.578	0.817	0.616	0.839	0.532	0.804	0.591	0.813
RBAN w/o Q2A	0.595	0.788	0.614	0.817	0.569	0.779	0.578	0.814	0.514	0.788	0.569	0.782
RBAN w/o A2Q	0.623	0.837	0.639	0.834	0.588	0.821	0.617	0.845	0.536	0.815	0.603	0.826
RBAN	**0.648**	**0.856**	**0.662**	**0.855**	**0.616**	**0.833**	**0.634**	**0.869**	**0.557**	**0.833**	**0.625**	**0.839**

Table 3: Performances of all the approaches to two sub-tasks, i.e., **Term-level** and **Category-level** ASC-QA. In each sub-task, all approaches are evaluated in three different domains, i.e., *Bags*, *Cosmetics* and *Electronics*.

approach without using question-to-answer attention. **10) RBAN w/o A2Q**. Our RBAN approach without using answer-to-question attention.

4.3 Experimental Results

Table 3 shows the performances of different approaches to ASC-QA. From this table, we can see that all the three state-of-the-art ASC approaches, i.e., **RAM**, **GCAE** and **S-LSTM**, perform better than **LSTM**. This confirms the usefulness of considering aspect information in ASC. Besides, both the attention based approaches **RAM** and **S-LSTM** achieve comparable or better performance than **GCAE**. This result demonstrates the usefulness of a proper attention mechanism to model aspect information.

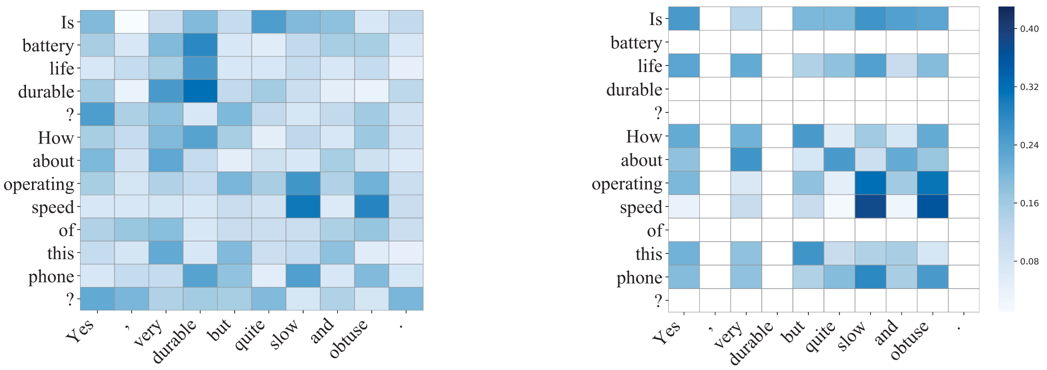
The two QA matching approaches, i.e., **BIDAF** and **HMN** could achieve comparable performance with the three state-of-the-art ASC approaches, and **MAMC** even beats all of them. This indicates the appropriateness of treating question and answer in a QA style review as two parallel units instead of a single sequence in ASC-QA.

Furthermore, our **RBAN w/o RAWS** approach (i.e., without considering aspect information) performs consistently better than **MAMC**. This encourages to employ bidirectional attention to learn the representation vectors of both the question and answer in order to capture the sentiment information therein. Besides, it's interesting to notice that **RBAN w/o A2Q** (i.e., without question vector s^q) performs much better than **RBAN w/o Q2A** (i.e., without answer vector s^a). This is due to the fact that the main sentiment polarity towards aspect is usually expressed in the answer text.

In comparison, when using RAWS, **RBAN** performs best and significantly outperforms **RBAN w/o RAWS** (p-value < 0.05), which encourages to discard noisy words for a specific aspect in both the question and answer sides. Impressively, in the sub-task of **Term-level ASC-QA**, compared to **LSTM**, **RBAN** achieves average improvements of 7.97% (F1) and 8.67% (Acc.) in three domains. In the sub-task of **Category-level ASC-QA**, compared to **LSTM**, **RBAN** achieves average improvements of 9.1% (F1) and 9.23% (Acc.). Significance test shows that these improvements are all significant (p-value < 0.05). These results encourage to incorporate both RAWS and bidirectional attentions to tackle ASC-QA.

5 Analysis and Discussion

Case Study. We provide a qualitative analysis of our approach on the development set. Specifically, in Figure 4, we visualize the attention matrix S^r in RBAN towards aspect "*operating speed*" in two cases, i.e., not using RAWS (Figure 4(a)) and using RAWS (Figure 4(b)). In Figure 4(a), color blue denotes attention weight (the darker the more important), we can find that both aspect "*battery life*" and aspect "*operating speed*" in question have been successfully matched with their corresponding answer phrases, i.e., "*very durable*" and "*quite slow and obtuse*". However, **RBAN without RAWS** can't discard noisy words (e.g., "*battery life*", "*durable*") for aspect "*operating speed*". In Figure 4(b), color white denotes the word inside question or answer has been discarded, we can find that **RBAN** is capable of effectively discarding noisy words such as "*battery*" and "*durable*" and highlighting those significant words such as "*slow*" and "*obtuse*" for aspect "*operating speed*".



(a) RBAN without RAWS

(b) RBAN with RAWS

Figure 4: Attention matrices for a QA text pair (each row is a question word and each column is an answer word). (a) and (b) show attention matrices of **RBAN without RAWS** and **RBAN** towards aspect term ***"operating speed"***.

Error Analysis. We randomly analyze 100 error cases in the experiments, which can be roughly categorized into 5 types. **1)** 27% errors are because that the answer length is too short. An example is "Question: *Is the screen good?* Answer: *No.*". **2)** 24% errors are due to negation words. An example is "*the case is not good*". Our approach fails to select the word "*not*" and incorrectly predicts *positive* polarity. This inspires us to optimize our approach so as to capture the negation scope better in the future. **3)** 19% errors are due to the wrong prediction on recognizing neutral instances. The shortage of neutral training examples makes the prediction of neutral instances very difficult. **4)** 16% errors are due to comparative opinions. An example is "*macos is much better than Windows*". Our approach incorrectly predicts *positive* for aspect "*Windows*". **5)** Finally, 14% errors are due to mistakes during Chinese word segmentation. An example is "好难看(very ugly)". It's incorrectly segmented into "好(good)|难(hard)|看(look)" and predicted as *positive*. This encourages to improve the performance of word segmentation on informal customer reviews.

6 Related Work

Existing studies on Aspect Sentiment Classification (ASC) could be divided into two groups according to the different level of text, i.e., sentence-level ASC and document-level ASC.

Sentence-level ASC is typically regarded as a sentence-level text classification which aims to incorporate aspect information into a model. Recently, Wang et al. (2016); Ma et al. (2017) propose an attention based LSTM to ASC by exploring the connection between an aspect and the content of a sentence. Tang et al. (2016b), Chen et al. (2017) and Wang et al. (2018b) employ memory networks to model the context and aspect. Wang and Lu (2018) propose a segmentation attention to capture structural dependency between target and opinion terms.

Document-level ASC aims to predict sentiment ratings for aspects inside a long text. Traditional studies (Titov and McDonald, 2008; Wang et al., 2010; Pontiki et al., 2016) solve document-level ASC as a sub-problem by utilizing heuristic based methods or topic models. Recently, Lei et al. (2016) focus on extracting rationales for aspects in a document. Li et al. (2018) propose an user-aware attention approach to document-level ASC. Yin et al. (2017) model document-level ASC as a machine comprehension problem, of which the input is also a parallel unit, i.e., question and answer. However, their question texts are pseudo and artificially constructed. This disaccords with the fact that real-world question texts also possibly involve multi-aspect and sentiment information.

Unlike all the above studies, this paper performs ASC on a different type of text, i.e., QA style reviews. To the best of our knowledge, this is the first attempt to perform ASC on QA style reviews.

7 Conclusion

In this paper, we propose a new task, i.e., Aspect Sentiment Classification towards Question Answering (ASC-QA). Specifically, we first build a high-quality human annotated benchmark corpus. Then, we design a reinforced bidirectional attention network (RBAN) approach to address ASC-QA. Empirical studies show that our proposed approach significantly outperforms several state-of-the-art baselines in the task of ASC-QA. In our future work, we would like to solve other challenges in ASC-QA such as data imbalance and negation detection to improve the performance. Furthermore, we would like to explore the effectiveness of our approach to ASC-QA in other languages.

Acknowledgments

We thank our anonymous reviewers for their helpful comments. This work was supported by three NSFC grants, i.e., No.61672366, No.61702149 and No.61525205. This work was also supported by the joint research project of Alibaba Group and Soochow University.

References

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of EMNLP-2017*, pages 452–461.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of ACL-2014*, pages 49–54.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of ACL-2011*, pages 151–160.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of EMNLP-2016*, pages 107–117.

Junjie Li, Haitong Yang, and Chengqing Zong. 2018. Document-level multi-aspect sentiment classification by jointly modeling users, aspects, and overall ratings. In *Proceedings of COLING-2018*, pages 925–936.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of IJCAI-2017*, pages 4068–4074.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proceedings of EMNLP-2013*, pages 1643–1654.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia V. Loukachevitch, Evgeniy V. Kotelnikov, Núria Bel, Salud María Jiménez Zafra, and Gülsen Eryigit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of NAACL-2016*, pages 19–30.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of NAACL-2015*, pages 486–495.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of COLING-2014*, pages 27–35.

Xipeng Qiu, Qi Zhang, and Xuanjing Huang. 2013. Fudannlp: A toolkit for chinese natural language processing. In *Proceedings of ACL-2013*, pages 49–54.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603.

Chenlin Shen, Changlong Sun, Jingjing Wang, Yangyang Kang, Shoushan Li, Xiaozhong Liu, Luo Si, Min Zhang, and Guodong Zhou. 2018a. Sentiment classification towards question-answering with hierarchical matching network. In *Proceedings of EMNLP-2018*, pages 3654–3663.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Sen Wang, and Chengqi Zhang. 2018b. Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling. In *Proceedings of IJCAI-2018*, pages 4345–4352.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of NIPS-1999*, pages 1057–1063.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect level sentiment classification with deep memory network. In *Proceedings of EMNLP-2016*, pages 214–224.

Ivan Titov and Ryan T. McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-2008*, pages 308–316.

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of IJCAI-2015*, pages 1347–1353.

Bailin Wang and Wei Lu. 2018. Learning latent opinions for aspect-level sentiment classification. In *Proceedings of AAAI-2018*, pages 5537–5544.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of SIGKDD-2010*, pages 783–792.

Jingjing Wang, Jie Li, Shoushan Li, Yangyang Kang, Min Zhang, Luo Si, and Guodong Zhou. 2018a. Aspect sentiment classification with both word-level and clause-level attention networks. In *Proceedings of IJCAI-2018*, pages 4439–4445.

Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. 2018b. Target-sensitive memory networks for aspect sentiment classification. In *Proceedings of ACL-2018*, pages 957–967.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of EMNLP-2016*, pages 606–615.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of ICML-2015*, pages 2048–2057.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of ACL-2018*, pages 2514–2523.

Yiming Yang and Xin Liu. 1999. A re-examination of text categorization methods. In *Proceedings of SIGIR-1999*, pages 42–49.

Yichun Yin, Yangqiu Song, and Ming Zhang. 2017. Document-level multi-aspect sentiment classification as machine comprehension. In *Proceedings of EMNLP-2017*, pages 2044–2054.