# Sentiment Classification towards Question-Answering with Hierarchical Matching Network

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

## Abstract

In an e-commerce environment, user-oriented question-answering (QA) text pair could carry rich sentiment information. In this study, we propose a novel task/method to address QA sentiment analysis. In particular, we create a high-quality annotated corpus with specially-designed annotation guidelines for QA-style sentiment classification. On the basis, we propose a three-stage hierarchical matching network to explore deep sentiment information in a QA text pair. First, we segment both the question and answer text into sentences and construct a number of [Q-sentence, A-sentence] units in each QA text pair. Then, by leveraging a QA bidirectional matching layer, the proposed approach can learn the matching vectors of each [Q-sentence, A-sentence] unit. Finally, we characterize the importance of the generated matching vectors via a self-matching attention layer. Experimental results, comparing with a number of state-of-the-art baselines, demonstrate the impressive effectiveness of the proposed approach for QA-style sentiment classification.

## 1 Introduction

Sentiment analysis, a.k.a. opinion mining, is a task which aims to identify the user sentiment orientation of a product/brand/service by monitoring the online textual data, e.g., reviews and social media messages. It has attracted huge attention in both academic and industrial communities due to its widespread applications, such like recommendation (Zhang et al., 2014) and social media mining (Chambers et al., 2015). As the fundamental component in sentiment analysis, sentiment classification mainly classifies the sentiment polarity as *positive* or *negative*, and has been well-studied from both sentence-level (Kim and Hovy, 2004) and document-level (Xu et al., 2016).

* Corresponding author

**Question 1:** *Is the screen clear? How is the battery?*
**Answer 1:** *It's a nice phone with high-quality screen. But the battery is not durable.*

**Question 2:** *Summer is coming, I'm afraid of getting darker. Is the sun cream really effective?*
**Answer 2:** *No, just depending on my own experience.*

Figure 1: Two examples of QA text pairs from "*customer questions & answers*" section in *Amazon*.

Recently, a new QA-style reviewing form, namely "*customer questions & answers*", has become increasingly popular on the giant e-commerce platforms, e.g., *Amazon* and *Taobao*. In this new form, a potential customer asks question(s) about the target product/service while other experienced user(s) can provide answer(s). With the widespread of such QA-style reviews, users find a different channel to efficiently explore rich and useful information, and service providers and scholars are paying more attention to its specific characteristics comparing with traditional reviews (Wachsmuth et al., 2014; Zhou et al., 2015a). Comparing to the traditional reviews, the QA style reviews can be more informative and convincing. More importantly, because answer providers are randomly picked from the users who already purchased the target item, this new form of review can be more reliable and trustful.

Regarding QA-style sentiment analysis, one straightforward method is to directly employ an existing sentiment classification approach that works well on traditional reviews, such as RNN (Nguyen and Shirai, 2015) and LSTM (Chen et al., 2016). However, because of the significant differences between QA-style and classical reviews, existing review mining algorithms, e.g., text-based sentiment analysis/classification, should not be di-


*Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3654–3663
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

rectly applied to this new kind of QA-style data. More detailed reasons can be found as the followings.

First, in QA-style text, the question and answer text are more likely to be two parallel units rather than a sequence form. On the one hand, for instance, in Figure 1, sentence "*It's a nice phone with high-quality screen.*" in **Answer 1** actually does not follow sentence "*How is the battery?*" in **Question 1** , but corresponds to sentence "*Is the screen clear?*" in **Question 1**. Therefore, when the question text and answer text are presented as two units in a sequence, it is rather difficult to capture the relationship between the question and its corresponding answer due to the possible long distance between them. On the other hand, there often exists both *positive* and *negative* sentiments in answer text according to different parts of question, and this specific case should be categorized as another category named *conflict*. For instance, in Figure 1, **Answer 1** "*It's a nice phone with high-quality screen. But the battery is not durable.*" is a *conflict* answer to **Question 1**. However, when this answer text is considered as a sequence, it is highly possible to be predicted as the category of *positive* or *negative* rather than *conflict*. In order to address these problems, a more appropriate approach is to segment both the question and answer text into some parallel sentences, and then construct the [Q-sentence, A-sentence] units in each QA text pair to detect in-depth sentiment information.

Second, although the main sentiment polarity is usually expressed from the answer text, the question text could also carry important sentiment tips to predict the sentiment polarity of a QA text pair. For instance, in Figure 1, we could hardly estimate the sentiment polarity solely based on **Answer 2**. However, when we take **Question 2**, "*Is the sun cream really effective?*", into consideration, it can be easier to label this QA text pair with a *negative* tag. In this study, we propose an approach to match the sentences inside the question and answer text bidirectionally.

Third, in each QA text pair, the importance degrees of different [Q-sentence, A-sentence] units can be different. For instance, in Figure 1, the [Q-sentence, A-sentence] unit, i.e., sentence "*Summer is coming, I'm afraid of getting darker.*" in **Answer 2** and sentence "*No, just depending on my own experience.*" in **Question 2**, makes tiny contribution to imply the sentiment polarity for the QA text pair. Therefore, a well-behaved network approach should consider the importance degrees of different [Q-sentence, A-sentence] units for predicting the sentiment polarity of a QA text pair.

The contribution of this paper is twofold. First, we propose a novel problem, QA-style sentiment analysis, and build a large-scale annotated corpus tailed for this task. The dataset is released to motivate future investigations for this track of research. Second, we propose a hierarchical matching network model to address the challenges of QA-style sentiment classification. Specifically, we first segment both the question and answer text into sentences and construct the [Q-sentence, A-sentence] units for each QA text pair. Then, by using a QA bidirectional matching layer, we encode each [Q-sentence, A-sentence] unit for exploring sentiment information. Finally, the self-matching attention layer in the model can capture the importance of these [Q-sentence, A-sentence] matching vectors obtained from QA bidirectional matching layer, which could effectively refine the evidence for inferring the sentiment polarity of a QA text pair. Experimental results show that the proposed approach significantly outperforms several strong baselines for QA-style sentiment classification.

## 2 Related Work

Sentiment classification has become a hot research field in NLP since the pioneering work by Pang et al. (2002). In general, the research on traditional sentiment classification has been carried out in different text levels, such like word-level, document-level and aspect-level.

Word-level sentiment classification has been studied in a long period in the research community of sentiment analysis. Some early studies have devoted their efforts to predicting the sentiment polarity of a word with different learning models and resources. Turney (2002) proposed an approach to predicting the sentiment polarity of words by calculating Pointwise Mutual Information (PMI) values between the seed words and the search hits. Hassan and Radev (2010) and Hassan et al. (2011) applied a Markov random walk model to determine the word polarities with a large word relatedness graph, and the synonyms and hypernyms in WordNet (Miller, 1995). More recently, some studies aim to learn better word embedding of a word rather than its polarity. Tang et al. (2014) developed three neural networks to learn word em-

| | *Positive* | *Negative* | *Conflict* | *Neutral* | *Total* |
|---|---|---|---|---|---|
| *Beauty* | 3,676 | 981 | 318 | 5,025 | 10,000 |
| *Shoe* | 4,025 | 819 | 412 | 4,744 | 10,000 |
| *Electronic* | 3,807 | 1,017 | 528 | 4,648 | 10,000 |

Table 1: Category distribution of the annotated data in three domains.

bedding by incorporating sentiment polarities of text in loss functions. Zhou et al. (2015b) employed both unsupervised and supervised neural networks to learn bilingual sentiment word embedding.

Document-level sentiment classification has also been studied in a long period in the research community of sentiment analysis. On one hand, many early studies have been devoted their efforts to various of aspects on learning approaches, such as supervised learning (Pang et al., 2002; Riloff et al., 2006), semi-supervised learning (Li et al., 2010; Xia et al., 2015; Li et al., 2015), and domain adaptation (Blitzer et al., 2007; He et al., 2011). On the other hand, many recent studies employ deep learning approaches to enhance the performances in sentiment classification. Tang et al. (2015) proposed a user-product neural network to incorporate both user and product information for sentiment classification. Xu et al. (2016) proposed a Cached Long Short-Term Memory neural networks (CLSTM) to capture the overall semantic information in long texts. More recently, Long et al. (2017) proposed a novel attention model, namely cognition-based attention, for sentiment classification.

Aspect-level sentiment classification is a relatively new research area in the research community of sentiment analysis and it is a fine-grained classification task. Recently, Wang et al. (2016) proposed an attention-based LSTM neural network to aspect-level sentiment classification by exploring the connection between an aspect and the content of a sentence. Tang et al. (2016) proposed a deep memory network with multiple attention-based computational layers to improve the performance. Wang et al. (2018) proposed a hierarchical attention network to explore both word-level and clause-level sentiment information towards a target aspect.

Unlike all the prior studies, this paper focuses on a very different kind of text representation, i.e., QA-style text level, for sentiment classification. To the best of our knowledge, this is the first attempt to perform sentiment classification on this text level.

## 3 Data Collection and Annotation

We collect QA text pairs from "*Asking All*" in *Taobao* (*Alibaba*)[1], which is the world's biggest e-commerce company. The QA text pairs are mainly from *Beauty*, *Shoe* and *Electronic* domains and each domain contains 10,000 QA text pairs.

We define four sentiment-related categories, i.e., *positive*, *negative*, *conflict* (both *positive* and *negative* sentiment) and *neutral* (neither *positive* nor *negative* sentiment). To guarantee a high annotation agreement, we propose some annotation guidelines after several times of annotation processes on a small size of data. Then, we ask more coders to annotate the whole data set according to these annotation guidelines.

The annotation guidelines contain two main groups. One contains the guidelines which aim to distinguish the categories of *neutral* and *non-neutral*, i.e.,

**(a)** A QA text pair in which the question and the answer do not match is annotated as a *neutral* sample. In this type of samples, the answer does not reply to the question correctly. **E1** is an example of this type where the question talks about the screen while the answer talks about the battery.

**E1:** Q: *Is the screen clear?*
A: *The battery life is decent.*

**(b)** A QA text pair with an unknown or uncertain answer is annotated as a *neutral* sample. **E2** is an example of this type.

**E2:** Q: *What about these sneakers?*
A: *I don't know, I bought it for my dad.*

**(c)** A QA text pair with only objective description is annotated as a *neutral* sample. **E3** is an example of this type.

**E3:** Q: *What's the operation system of the phone?*
A: *Android.*

**(d)** A QA text pair which compares two different products is annotated as a *neutral* sample. In this type of samples, two products are involved and it

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

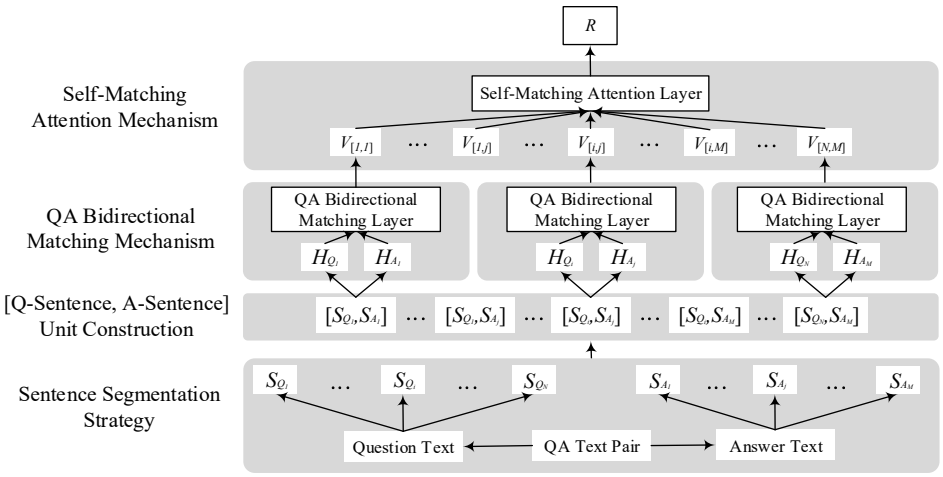


Figure 2: The overview of our approach to QA-style sentiment classification where $S_{Q_i}$ denotes the $i$-th sentence in question text, $S_{A_j}$ denotes the $j$-th sentence in answer text, $H_{Q_i}$ and $H_{A_j}$ denote the contextual representations for $S_{Q_i}$ and $S_{A_j}$ respectively, $V_{[i,j]}$ denotes the bidirectional matching vector for $[S_{Q_i}, S_{A_j}]$ unit through QA bidirectional matching layer, and $R$ is the QA text pair representation refined by self-matching attention layer.

is sometimes difficult to tell the sentiment orientation of one product. **E4** is an example of this type.

**E4:** Q: *How about this phone when compared to iPhone 6s?*
A: *It's up to you, and they're not comparable.*

The other group contains the guidelines which aim to distinguish the categories of *positive* and *negative*, i.e.,
**(e)** If the answer text contains sentimental expressions to question like "*disappointed*", "*terrible*", and so on, we annotate it as *negative*. **E5** is an example of this type.

**E5:** Q: *How is the rock climbing shoe?*
A: *I am so disappointed, my feet felt hurt when I wore them.*

**(f)** If the answer text contains sentimental expressions to question like "*perfect*", "*satisfied*", and so on, we annotate it as *positive*. **E6** is an example of this type.

**E6:** Q: *How about the fragrance?*
A: *I am so satisfied, it smells distinctive.*

**(g)** If we cannot confirm the polarity of a QA text pair only depending on answer text, we annotate the polarity according to both the question and answer text. For instance, **E7** is an example with *positive* polarity, while **E8** is an example with *negative* polarity.

**E7:** Q: *Will the phone get hot when gaming?*
A: *No.*

**E8:** Q: *Is the sun cream really economic?*
A: *No.*

We assign two annotators to annotate each QA text pair, and the *Kappa* consistency check value of the annotation is 0.84. When annotators cannot reach an agreement, an expert will make the final decision, ensuring the quality of data annotation. Table 1 shows the category distribution of the corpus. To motivate other scholars to investigate this novel but important task, we share the data via Github[2].

## 4 Methodology

In this section, we introduce the proposed hierarchical matching network approach for QA-style sentiment classification. Figure 2 depicts the overview of the proposed approach.

### 4.1 QA Bidirectional Matching Mechanism

**Word Encoding Layer:** After sentence segmentation, the question text in a QA text pair contains $N$ sentences, $S_{Q_i}$ represents the $i$-th sentence in the question text. Similarly, the answer text in this QA text pair contains $M$ sentences, $S_{A_j}$ represents the $j$-th sentence in the answer text. We then construct [Q-sentence, A-sentence] units by pairing one sentence in the question text and one sentence in the answer text, and we obtain $N*M$ [Q-sentence, A-sentence] units at last.

Given a $[S_{Q_i}, S_{A_j}]$ unit in this QA text pair, i.e., Q-sentence $S_{Q_i}$ with words $w_{i,n}, i \in [1, N], n \in$

[2]https://github.com/clshenNLP/QASC/

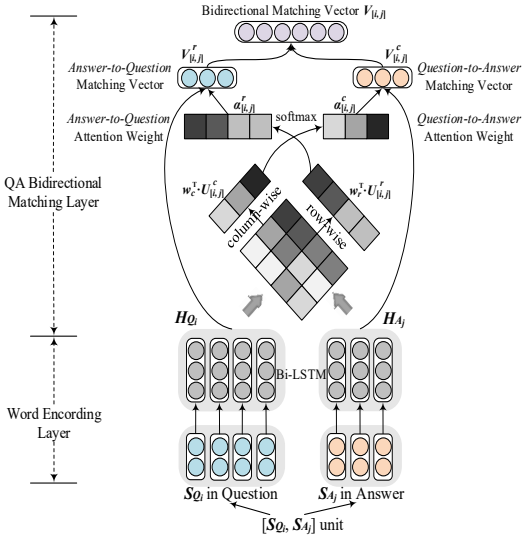


Figure 3: The detail architecture of QA bidirectional matching mechanism.

$[1, N_i]$ and A-sentence $S_{A_j}$ with words $w_{j,m}, j \in [1, M], m \in [1, M_j]$, we first convert the words to their respective word embeddings ($x_{i,n} \in \mathbb{R}^d, i \in [1, N], n \in [1, N_i]$ and $x_{j,m}, j \in [1, M], m \in [1, M_j]$). We then use Bi-directional LSTM (namely Bi-LSTM), which can efficiently make use of past features (via forward states) and future features (via backward states) for a specific time step, to get contextual representations of $S_{Q_i}$ and $S_{A_j}$ individually. The representation of each word is formed by concatenating the forward and backward hidden states. For simplicity, we note contextual representation of $S_{Q_i}$ as $H_{Q_i}$ , and contextual representation of $S_{A_j}$ as $H_{A_j}$ respectively:

$$H_{Q_i} = [h_{i,1}, h_{i,2}, ..., h_{i,n}, ..., h_{i,N_i}] \quad (1)$$

$$H_{A_j} = [h_{j,1}, h_{j,2}, ..., h_{j,m}, ..., h_{j,M_j}] \quad (2)$$

where $h_{i,n} \in \mathbb{R}^{d'}$ denotes the word representation in $S_{Q_i}$ at time step $n$, $h_{j,m} \in \mathbb{R}^{d'}$ denotes the word representation in $S_{A_j}$ at time step $m$, and $d'$ is the dimensionality of word representation.

**QA Bidirectional Matching Layer:** General neural network could not capture sentiment matching information in a $[S_{Q_i}, S_{A_j}]$ unit well. For the sake of solving this problem, we introduce the QA bidirectional matching layer to encapsulate the clues and interactions between $S_{Q_i}$ and $S_{A_j}$ synchronously (Tay et al., 2017; McCann et al., 2017). Figure 3 depicts the detail architecture of QA bidirectional matching mechanism. Specifically, we first calculate the bidirectional pair-wise matching matrix by using the following formula:

$$D_{[i,j]} = (H_{Q_i})^\top \cdot (H_{A_j}) \quad (3)$$

where $D_{[i,j]} \in \mathbb{R}^{N_i \times M_j}$ denotes the bidirectional matching matrix for the $[S_{Q_i}, S_{A_j}]$ unit. Each element in $D_{[i,j]}$ is the score that measures how well the word in $S_{Q_i}$ semantically matches the word in $S_{A_j}$ and vice versa.

Given the bidirectional matching matrix $D_{[i,j]}$, we use attention mechanism (Yang et al., 2016; Cui et al., 2017) to mine the sentiment matching information between question and answer from two directions, which could be seen as an *Answer-to-Question* attention and a *Question-to-Answer* attention as follows.

• ***Answer-to-Question* Attention:** We employ row-wise operations to compute the attention weight vector $\alpha^r_{[i,j]}$ as follows:

$$U^r_{[i,j]} = \tanh(W_r \cdot D^\top_{[i,j]}) \quad (4)$$

$$\alpha^r_{[i,j]} = \text{softmax}(w_r^\top \cdot U^r_{[i,j]}) \quad (5)$$

where $\alpha^r_{[i,j]} \in \mathbb{R}^{N_i}$ is the *Answer-to-Question* attention weight vector regarding the importance degrees of all words in Q-sentence $S_{Q_i}$, $W_r \in \mathbb{R}^{d' \times M_j}$ and $w_r \in \mathbb{R}^{d'}$ are weight matrices. After computing the *Answer-to-Question* attention weight vector, we can get the *Answer-to-Question* matching vector $V^r_{[i,j]} \in \mathbb{R}^{d'}$ as follows:

$$V^r_{[i,j]} = (H_{Q_i}) \cdot \alpha^r_{[i,j]} \quad (6)$$

• ***Question-to-Answer* Attention:** Simultaneously, we employ column-wise operations to calculate the attention weight vector $\alpha^c_{[i,j]}$ as follows:

$$U^c_{[i,j]} = \tanh(W_c \cdot D_{[i,j]}) \quad (7)$$

$$\alpha^c_{[i,j]} = \text{softmax}(w_c^\top \cdot U^c_{[i,j]}) \quad (8)$$

where $\alpha^c_{[i,j]} \in \mathbb{R}^{M_j}$ is the *Question-to-Answer* attention weight vector regarding the importance degrees of all words in A-sentence $S_{A_j}$, $W_c \in \mathbb{R}^{d' \times N_i}$ and $w_c \in \mathbb{R}^{d'}$ are weight matrices. After calculating the *Question-to-Answer* attention weight vector, we can get the *Question-to-Answer* matching vector $V^c_{[i,j]} \in \mathbb{R}^{d'}$ as follows:

$$V^c_{[i,j]} = (H_{A_j}) \cdot \alpha^c_{[i,j]} \quad (9)$$

Then, we combine *Answer-to-Question* and *Question-to-Answer* matching vectors to represent

the final bidirectional matching vector of the $[S_{Q_i}, S_{A_j}]$ unit:

$$V_{[i,j]} = V^r_{[i,j]} \oplus V^c_{[i,j]} \quad (10)$$

where $\oplus$ denotes the concatenate operator, and $V_{[i,j]}$ denotes the bidirectional matching vector which integrates $S_{Q_i}$ and $S_{A_j}$ with each other.

### 4.2 Self-Matching Attention Mechanism

Through the QA bidirectional matching layer, informative bidirectional matching vectors are generated to pinpoint the sentiment matching information in each [Q-sentence, A-sentence] unit. Intuitively, each matching vector for [Q-sentence, A-sentence] unit holds different importance to a QA text pair. To better aggregate the evidence from these vectors for inferring the sentiment polarity of the QA text pair, we propose a self-matching attention layer, matching these informative vectors against themselves.

**Self-Matching Attention Layer**: As aforementioned, we have obtained $N*M$ bidirectional matching vectors through QA bidirectional matching layer, then we calculate the attention weight vector $\alpha$ with these matching vectors by following formulas:

$$V = [V_{[1,1]}, V_{[1,2]}, ..., V_{[i,j]}, ..., V_{[N,M]}] \quad (11)$$

$$U = \tanh(W_h \cdot V) \quad (12)$$

$$\alpha = \mathrm{softmax}(w_h^\top \cdot U) \quad (13)$$

where $\alpha$ is the attention weight vector which measures the importance of these matching vectors, $W_h$ and $w_h$ are the weight matrices.

Finally, we can get the QA text pair representation $R$ as follows:

$$R = V \cdot \alpha \quad (14)$$

### 4.3 Classification Model

QA text pair representation $R$ is a high level representation which can be used for classification. In our approach, we feed $R$ to a *softmax* classifier:

$$p = \mathrm{softmax}(W_l \cdot R + b_l) \quad (15)$$

where $p$ is a set of predicted distribution of the sentiment categories, i.e., *positive*, *negative*, *neutral*, and *conflict*. $W_l$ is the weight matrix and $b_l$ is the bias.

To learn the whole model, we train an end-to-end model given the training data, and the goal of training is to minimize the cross-entropy loss, i.e.,

$$L(\theta) = -\sum_{s=1}^{S}\sum_{k=1}^{K} y_s^k \cdot \log\hat{y}_s^k + \lambda\|\theta\|_2^2 \quad (16)$$

where $S$ is the number of training data. $y_s$ is the true sentiment label of the $s$-th sample. $\hat{y}_s$ is the predicted sentiment label of the $s$-th sample. $K$ is number of all sentiment categories. $\lambda$ is a $L_2$-regularization term, $\theta$ is the parameter set. In the above equation, the model parameters are optimized by using Adam (Kingma and Ba, 2014).

## 5 Experimentation

In this section, we evaluate the performances of the proposed approach for QA-style sentiment classification.

### 5.1 Experimental Settings

• **Data Sets:** As introduced in Section 3, the annotated QA text pairs cover three different domains. In each domain, we randomly split the data into a training set (80% in each category) and a test set (20% in each category). In addition, we set aside 10% from the training set as the development data for parameters tuning.

• **Word Segmentation and Embeddings:** FudanNLP[3] (Qiu et al., 2013) is employed to segment text into Chinese words and word2vec[4] (Mikolov et al., 2013) is employed to pre-train word embeddings. The vector dimensionality is set to be 100.

• **Sentence Segmentation:** CoreNLP[5] (Manning et al., 2014) is employed to segment both the question and answer text into sentences.

• **Hyper-parameters:** In the experiment, all out-of-vocabulary words are initialized by sampling from the uniform distribution $U(-0.01, 0.01)$. All weight matrices are given their initial values by sampling from uniform distribution $U(-0.01, 0.01)$. The LSTM hidden states are set to be 128 and all models are trained by mini-batch of 32 instances. The dropout rate is set to 0.2. The other hyper-parameters are tuned according to the development data.

• **Evaluation Metric:** The performance is evaluated using standard *Accuracy* and *Macro-F1*.

[3]https://github.com/FudanNLP/fnlp/

[4]https://code.google.com/archive/p/word2vec/

[5]http://stanfordnlp.github.io/CoreNLP/

| | *Beauty* | | *Shoe* | | *Electronic* | |
|---|---|---|---|---|---|---|
| | *Macro-F1* | *Accuracy* | *Macro-F1* | *Accuracy* | *Macro-F1* | *Accuracy* |
| SVM | 0.362 | 0.684 | 0.381 | 0.718 | 0.435 | 0.691 |
| LSTM | 0.499 | 0.712 | 0.520 | 0.754 | 0.562 | 0.715 |
| Bi-LSTM | 0.527 | 0.719 | 0.531 | 0.759 | 0.574 | 0.723 |
| Bidirectional-Match | 0.526 | 0.747 | 0.557 | 0.796 | 0.582 | 0.741 |
| AtoQ-Match | 0.543 | 0.745 | 0.602 | 0.792 | 0.567 | 0.754 |
| QtoA-Match | 0.573 | 0.751 | 0.647 | 0.807 | 0.608 | 0.752 |
| Bidirectional-Match QA | 0.583 | 0.760 | 0.666 | 0.815 | 0.617 | 0.764 |
| HMN | **0.598** | **0.776** | **0.683** | **0.827** | **0.640** | **0.779** |

Table 2: Performance of our approaches to QA-style sentiment classification in all domains.

### 5.2 Experimental Results

The following baseline approaches are employed for comparison. Note that all the approaches share the same word embeddings for fair comparison.

• **SVM:** This baseline employs support vector machine along with word embedding features. The question and answer text in a QA text pair are chained as a sequence.

• **LSTM:** A standard LSTM model utilizes word embeddings and concatenates the question and answer text as a sequence.

• **Bi-LSTM:** A bidirectional LSTM model which concatenates the question and answer text as a sequence.

• **Bidirectional-Match:** This approach employs QA bidirectional matching mechanism, without taking the sentence segmentation strategy and self-matching attention mechanism.

• **AtoQ-Match:** This approach takes the sentence segmentation strategy, and employs QA unidirectional matching mechanism (i.e., only using *Answer-to-Question* attention), but does not employ self-matching attention mechanism. We average the *Answer-to-Question* matching vectors to represent the QA text pair.

• **QtoA-Match:** This approach takes the sentence segmentation strategy, and employs QA unidirectional matching mechanism (i.e., only using *Question-to-Answer* attention), but does not employ self-matching attention mechanism.

• **Bidirectional-Match QA:** This approach takes the sentence segmentation strategy, and employs QA bidirectional matching mechanism, but does not employ self-matching attention mechanism.

• **HMN:** This is our hierarchical matching network model which takes the sentence segmentation strategy and employs both QA bidirectional matching mechanism and self-matching attention mechanism.

Table 2 summarizes the experimental results of all the approaches above, and we can find that:

(1) All LSTM-based approaches are superior to **SVM**, indicating the effectiveness of neural network for this task.

(2) The proposed approaches, with novel QA contextual representation, outperform the other baseline approaches.

(3) When only employing QA bidirectional matching mechanism, **Bidirectional-Match QA**, which takes the sentence segmentation strategy, consistently outperforms **Bidirectional-Match** (without sentence segmentation) in all domains. It confirms our hypothesis that sentence segmentation helps to extract the sentiment matching information between the question and answer.

(4) When comparing to QA unidirectional matching mechanism, **Bidirectional-Match QA**, which employs QA bidirectional matching mechanism, performs better than **AtoQ-Match** and **QtoA-Match**. It confirms our hypothesis that both the question and answer information contribute to sentiment polarity of the QA text pair.

(5) Impressively, the proposed approach **HMN** significantly outperforms all the other approaches in all domains ($p$-value$<$0.05 via $t$-test). It verifies the advantages of both QA bidirectional matching mechanism and self-matching attention mechanism for this task.

Besides, we also implement some more recent state-of-the-art approaches for sentiment classification, which are illustrated in Table 3. This result also supports the earlier findings.

• **CNN-Tensor** (Lei et al., 2015)**:** This is a state-of-the-art approach to sentence-level sentiment classification, which models $n$-gram interactions based on tensor product and evaluates all non-

| | *Beauty* | | *Shoe* | | *Electronic* | |
|---|---|---|---|---|---|---|
| | *Macro-F1* | *Accuracy* | *Macro-F1* | *Accuracy* | *Macro-F1* | *Accuracy* |
| CNN-Tensor | 0.500 | 0.731 | 0.535 | 0.765 | 0.576 | 0.734 |
| Attention-LSTM | 0.509 | 0.725 | 0.571 | 0.755 | 0.576 | 0.721 |
| BiMPM | 0.553 | 0.745 | 0.587 | 0.766 | 0.584 | 0.746 |
| HMN | **0.598** | **0.776** | **0.683** | **0.827** | **0.640** | **0.779** |

Table 3: The proposed approach vs. several strong baseline approaches in all domains.

| **E9:** Domain: *Beauty* True Label: *neutral* | | | |
|---|---|---|---|
| **Q**: 请问各位朋友们，淡斑效果怎么样？谢谢！<br>(*Hey, friends, how about the spot-fading of this product? Thanks a lot!*)<br>**A**: 讲真哦，保湿还不错！<br>(*To tell you the truth, it can moisturize effectively!*) | | | |
| CNN-Tensor | Attention-LSTM | BiMPM | HMN |
| ✗ (*positive*) | ✗ (*positive*) | ✓ (*neutral*) | ✓ (*neutral*) |
| **E10:** Domain: *Electronic* True Label: *conflict* | | | |
| **Q**: 这款笔记本怎么样呢？电池耐用吗？玩大型游戏卡不卡？<br>(*How about this notebook? Is the battery durable? Does the OS run fast when playing games?*)<br>**A**: 电池不是很耐用。玩大型游戏有点卡。其他的都还好。<br>(*Battery isn't much durable. The OS doesn't run fast when playing games. The other aspects are good.*) | | | |
| CNN-Tensor | Attention-LSTM | BiMPM | HMN |
| ✗ (*positive*) | ✗ (*negative*) | ✗ (*negative*) | ✓ (*conflict*) |

Table 4: Some examples in the test data with their predicted categories by some approaches where ✗ (or ✓) means that the predicted category is wrong (or correct).

consecutive *n*-gram vectors as a feature mapping operator for CNNs.

• **Attention-LSTM** (Wang et al., 2016)**:** This is a state-of-the-art approach to aspect-level sentiment classification. In our implementation, we ignore the aspect embedding and directly use the outputs of LSTM to yield the attention.

• **BiMPM** (Wang et al., 2017)**:** This is a state-of-the-art approach to QA matching, which matches the question and answer from multiple perspectives. In our implementation, we use the matching representation to perform QA-style sentiment classification with a *softmax* classifier.

• **HMN:** The proposed hierarchical matching network which employs both QA bidirectional matching mechanism and self-matching attention mechanism, and takes the sentence segmentation strategy.

Table 3 shows the comparison results of these strong baseline approaches and the proposed approach (**HMN**) in all domains. From this table, we can find that: (1) the approaches that take matching strategy, i.e., **BiMPM** and our approach **(HMN)**, outperform other approaches. (2) The proposed approach **(HMN)** significantly outperforms all the other baseline approaches in terms of both *Macro-F1* and *Accuracy* ($p$-value$<$0.05 via $t$-test), which confirms the initial hypotheses of this study.

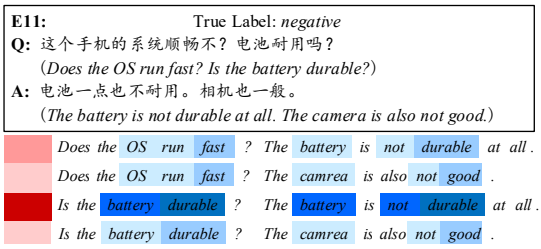


Figure 4: The attention visualizations for a QA text pair.

### 5.3 Case Study

Table 4 shows some examples, along with the predicted categories via different approaches. We can find that: (1) the approaches based on matching strategy (**BiMPM** and **HMN**) are well-performed, as shown in **E9**, when question and answer carrying different kinds of information. This is a unique challenge for QA-style sentiment mining, and traditional sentiment classification approaches can hardly address this problem. (2) The proposed approach **(HMN)** performs better than other approaches when dealing with *conflict* instances, as shown in **E10**.

### 5.4 Visualization of Attention

To get a better understanding of our proposed hierarchical matching network for QA-style sentiment classification, we picture the attention weights obtained from Equations (5), (8) and (13). For

simplicity, we directly use the English translation of **E11** for illustration and adopt the visualization approach presented by Yang et al. (2016), as shown in Figure 2. Specifically, each line is a [Q-sentence, A-sentence] unit, where the red denotes the [Q-sentence, A-sentence] unit weight, the blue denotes the word weight in each [Q-sentence, A-sentence], and the color depth indicates the importance of attention weights (the darker the more important).

From Figure 4, we can see that the QA bidirectional matching layer always assigns reasonable attention weights to words in each [Q-sentence, A-sentence] unit which makes sentence from question and sentence from answer match correctly. In addition, the self-matching attention layer is able to select informative [Q-sentence, A-sentence] unit for predicting true sentiment polarity of this example.

## 6 Conclusion

In this paper, we propose a novel but important sentiment analysis task, i.e., QA-style sentiment mining, and we build a large-scale high-quality human annotated corpus for experiment. The dataset is shared to encourage other scholars to investigate this interesting problem. Moreover, we propose a hierarchical matching neural network model to enable QA bidirectional matching mechanism and self-matching attention mechanism for this task. Empirical studies show that the proposed approach significantly outperforms other strong baseline approaches in all the test domains for QA-style sentiment classification.

In the future, we would like to investigate some other network structures to explore deeper information in each QA text pair. Besides, we would like to test the effectiveness of the proposed approach to QA-style sentiment classification in some other languages.

## Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work is partially supported by the National Key R&D Program of China under Grant No.2017YFB1002101 and two NSFC grants No.61331011, No.61672366. This work is also supported by the joint research project of Alibaba Group and Soochow University.

## References

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL-2007*, pages 440–447.

Nathanael Chambers, Victor Bowen, Ethan Genco, Xisen Tian, Eric Young, Ganesh Harihara, and Eugene Yang. 2015. Identifying political sentiment between nation states with social media. In *Proceedings of EMNLP-2015*, pages 65–75.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Enhancing and combining sequential and tree LSTM for natural language inference. *arXiv preprint arXiv:1609.06038*.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of ACL-2017*, pages 593–602.

Ahmed Hassan, Amjad Abu-Jbara, Rahul Jha, and Dragomir Radev. 2011. Identifying the semantic orientation of foreign words. In *Proceedings of ACL-2011*, pages 592–597.

Ahmed Hassan and Dragomir Radev. 2010. Identifying text polarity using random walks. In *Proceedings of ACL-2010*, pages 395–403.

Yulan He, Chenghua Lin, and Harith Alani. 2011. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of ACL-2011*, pages 123–131.

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of COLING-2004*, pages 1367–1374.

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

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2015. Molding CNNs for text: non-linear, non-consecutive convolutions. In *Proceedings of EMNLP-2015*, pages 1565–1575.

Shoushan Li, Chu-Ren Huang, Guodong Zhou, and Sophia Yat Mei Lee. 2010. Employing personal/impersonal views in supervised and semi-supervised sentiment classification. In *Proceedings of ACL-2010*, pages 414–423.

Shoushan Li, Lei Huang, Jingjing Wang, and Guodong Zhou. 2015. Semi-stacking for semi-supervised sentiment classification. In *Proceedings of ACL-IJCNLP-2015*, pages 27–31.

Yunfei Long, Lu Qin, Rong Xiang, Minglei Li, and Chu-Ren Huang. 2017. A cognition based attention model for sentiment analysis. In *Proceedings of EMNLP-2017*, pages 462–471.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL-2014*, pages 55–60.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Proceedings of NIPS-2017*, pages 6294–6305.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS-2013*, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Thien Hai Nguyen and Kiyoaki Shirai. 2015. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In *Proceedings of EMNLP-2015*, pages 2509–2514.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of ACL-2002*, pages 79–86.

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

Ellen Riloff, Siddharth Patwardhan, and Janyce Wiebe. 2006. Feature subsumption for opinion analysis. In *Proceedings of EMNLP-2006*, pages 440–448.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of ACL-IJCNLP-2015*, pages 1014–1023.

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

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of ACL-2014*, pages 1555–1565.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2017. A compare-propagate architecture with alignment factorization for natural language inference. *arXiv preprint arXiv:1801.00102*.

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL-2002*, pages 417–424.

Henning Wachsmuth, Martin Trenkmann, Benno Stein, and Gregor Engels. 2014. Modeling review argumentation for robust sentiment analysis. In *Proceedings of COLING-2014*, pages 553–564.

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

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

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of IJCAI-2017*, pages 4144–4150.

Rui Xia, Cheng Wang, Xin-Yu Dai, and Tao Li. 2015. Co-training for semi-supervised sentiment classification based on dual-view bags-of-words representation. In *Proceedings of ACL-IJCNLP-2015*, pages 1054–1063.

Jiacheng Xu, Danlu Chen, Xipeng Qiu, and Xuanjing Huang. 2016. Cached long short-term memory neural networks for document-level sentiment classification. In *Proceedings of EMNLP-2016*, pages 1660–1669.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT-2016*, pages 1480–1489.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of SIGIR-2014*, pages 83–92.

Guangyou Zhou, Tingting He, Jun Zhao, and Wensheng Wu. 2015a. A subspace learning framework for cross-lingual sentiment classification with partial parallel data. In *Proceedings of IJCAI-2015*, pages 1426–1433.

Huiwei Zhou, Long Chen, Fulin Shi, and Degen Huang. 2015b. Learning bilingual sentiment word embeddings for cross-language sentiment classification. In *Proceedings of ACL-IJCNLP-2015*, pages 430–440.