Question-answering Aspect Classification with Hierarchical Attention Network

Hanqian Wu[1,2], Mumu Liu[1,2], Jingjing Wang[3,⋆], Jue Xie[4], and Chenlin Shen[3]

[1] School of Computer Science and Engineering, Southeast University, China
[2] Key Laboratory of Computer Network and Information Integration of Ministry of Education, Southeast University, China
[3] NLP Lab, School of Computer Science and Technology, Soochow University, China
[4] Southeast University-Monash University Joint Graduate School, China
hanqian@seu.edu.cn, liudoublemu@163.com, djingwang@gmail.com

Abstract. In e-commerce websites, user-generated question-answering text pairs generally contain rich aspect information of products. In this paper, we address a new task, namely Question-answering (QA) aspect classification, which aims to automatically classify the aspect category of a given QA text pair. In particular, we build a high-quality annotated corpus with specifically designed annotation guidelines for QA aspect classification. On this basis, we propose a hierarchical attention network to address the specific challenges in this new task in three stages. Specifically, we firstly segment both question text and answer text into sentences, and then construct (sentence, sentence) units for each QA text pair. Second, we leverage a QA matching attention layer to encode these (sentence, sentence) units in order to capture the aspect matching information between the sentence inside question text and the sentence inside answer text. Finally, we leverage a self-matching attention layer to capture different importance degrees of different (sentence, sentence) units in each QA text pair. Experimental results demonstrate that our proposed hierarchical attention network outperforms some strong baselines for QA aspect classification.

Keywords: Aspect Classification · Question Answering · Hierarchical Attention

1 Introduction

Recently, a new question-answering (QA) style reviewing form, namely customer questions and answers, has been widely adopted in many popular e-commerce platforms, such as Amazon, Taobao and JingDong. Figure 1 gives an example of QA-style reviewing form. In this new reviewing form, a customer can ask questions about certain product which he/she wants to purchase while other customers who have bought this product can answer these questions. With the widespread of such QA-style reviewing form, the relevant research is drawing

⋆ Corresponding Author

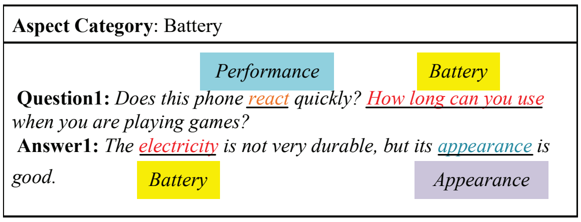


Fig. 1. A translated example of QA-style reviews from an e-commerce website.

more and more attention due to its specific characteristics. On one hand, users prefer to write QA-style reviews instead of traditional reviews. On the other hand, compared with the traditional product reviews, the QA-style reviewing form is more informative and convincing. This largely avoids fake reviews and makes product reviews more reliable. Thus, aspect-based sentiment analysis for QA-style texts becomes particularly important, in which aspect classification is a critically basic subtask. However, there are no existing studies with focus on aspect classification of QA-style reviews which aims to identify the aspect category of a given QA text pair. Further, according to the corpus analysis, this new task has the following specific characteristics.

First, from Figure 1, we could find that **Question1** involves two different aspects distributed in two different sentences, i.e., *performance* inside the first sentence and *battery* inside the second sentence, while **Answer1** involves one related aspect *battery* inside the first sentence and the other unrelated aspect *appearance* inside the second sentence. Under this circumstance, we only need to focus on identifying the matching aspect *battery* between **Question1** and **Answer1**, and ignore the unrelated and non-matching parts between **Question1** and **Answer1**. A better way to handle the above challenge is to firstly perform sentence segmentation in each QA text pair in order to segment both question and answer texts into sentences and make each sentence contain only one aspect. Then, after sentence segmentation, we construct the (sentence, sentence) units in each QA text pair to detect the in-depth aspect matching information, where each (sentence, sentence) unit is composed of two sentences from question and answer text respectively. Finally, in this study, we propose an innovative attention-based neural layer, which could match the sentences inside question and answer text, to encode the (sentence, sentence) units.

Second, for a specific aspect, the importance degree of different (sentence, sentence) units in each QA text pair may be different. For instance, the (sentence, sentence) unit, i.e., sentence "*Does this phone react quickly?*" in **Question1** and sentence "*but its appearance is good.*" in **Answer1**, fails to contribute in identifying aspect *battery*, while the (sentence, sentence) unit, i.e., sentence "*How long can you use when you are playing games?*" in **Question1** and sentence "*The electricity is not very durable,*" in **Answer1** contributes much in identifying aspect *battery*. Therefore, a well-behaved neural network approach should be capable of capturing the importance degree of different (sentence, sentence) units in each QA text pair.

To address the above specific characteristics in QA aspect classification, in this paper, we build a high-quality annotated corpus tailored for the new QA aspect classification task and then propose a hierarchical attention network to address the new task. Specifically, we firstly perform sentence segmentation in order to segment both the question and answer texts into sentences and then construct (sentence, sentence) units in each QA text pair. Then, we leverage an innovative QA matching attention layer to encode these (sentence, sentence) units in order to match the sentences inside question and answer texts. Finally, we leverage a self-matching attention layer to capture the different importance degrees of different (sentence, sentence) units in each QA text pair. Empirical studies demonstrate that our hierarchical attention approach performs better than several baseline approaches in the task of QA aspect classification.

2 Related Work

Aspect classification, also known as aspect category classification, is regarded as a special case of text classification and often treated as a supervised classification task. Thus, approaches to text classification, such as CNN[4], LSTM [12] and so on, can be applied to aspect classification task, but very few research has been conducted on aspect classification. Toh et al. [11] train binary classifiers based on sigmoidal feedforward network for aspect category classification. Xue et al. [14] propose a multi-task learning model based on neural networks to solve the two tasks aspect category classification and aspect term extraction together.

Besides, aspect classification task is also related to the aspect extraction task. Aspect extraction aims to extract the fine-grained opinion targets from reviews, which is divided into two subtasks, i.e., extracting all aspect terms from corpus and clustering aspect terms with similar meaning into categories where each category represents an aspect. Poria et al. [7] exploit common-sense knowledge and sentence dependency trees to detect both explicit and implicit aspects from opinionated texts. Rana et al. [8] propose a two-fold rule-based method based on sequential patterns and rules mined from reviews. Supervised learning, such as Conditional Random Fields(CRF), requires more manual annotation. Shu et al. [9] propose a lifelong CRF model for aspect extraction which leverages the knowledge from many past domains to facilitate extraction for a new domain. Besides, unsupervised methods, such as Latent Dirichlet Allocation(LDA) and its variants [6], have been applied to aspect extraction. To tackle the weakness of the LDA-based methods, He et al. [3] propose a neural approach based on attention mechanism to de-emphasize irrelevant words during training and further improve the coherence of aspects.

Unlike above studies, this paper focuses on the aspect classification task for QA text pairs. To the best of our knowledge, we are the first to focus on identifying the aspect category of a QA text pair.

3 Data Collection and Annotation

We manually annotate 8,313 QA text pairs collected from "*Asking All*" in Taobao[5] which is one of the most famous e-commerce platforms in China. The QA text pairs are mainly extracted from the domain of *electronic appliances*. To ensure the high consistency of corpus annotation, we define three aspect-related annotation guidelines and annotate each QA text pair in the form of a triple of *aspect term*, *aspect* and *polarity* (all examples presented in this paper are translations of original Chinese texts).

Guideline1 The annotation of a QA text pair with a triple depends on whether we can extract the aspect term from the question text.

Guideline2 All aspects, i.e., aspect categories, are induced from aspect terms in our corpus. For example, aspect terms "*electricity*" and "*durability*" can be categorized into the aspect *battery*. In other words, these aspect categories are predefined based on the annotation of aspect terms. Aspects are divided into two categories, one is domain-independent aspect, i.e., the aspect does not change with the migration of domain and is applied to all domains, such as *weight*, *quality* and *appearance*. The other one is domain-dependent aspect, i.e., the aspect is specific to the particular domain, such as *performance*, *battery* and *IO*.

Guideline3 If an aspect is contained in both the question and answer texts, we need to determine the sentiment polarity of this aspect so as to annotate this QA text pair with a triple. In general, the sentiment polarity can be divided into *neutral* and *non-neutral* categories. Further, the *non-neutral* category can be subdivided into *positive*, *negative* and *conflict*(a mix of both positive and negative) categories. Some cases may occur in the processing of annotation, elaborated as follows:

(a) If the answer text objectively describes the aspect in the question text, the QA text pair is annotated as (*aspect term*, *aspect*, *neutral*). **E1** is an example of this category. In the question text, aspect terms "*battery*" and "*communication effect*" belong to aspects *battery* and *IO* respectively, while the valid answer is only related to the aspect *IO* and objectively describes it.

E1: Q: How about the battery? How about the communication effect?
A: The ring is low, but its appearance is very good.

(b) If the answer text contains emotional words that express sentiment of the aspect in the question text, such as "*bad*" and "*not good*", the QA text pair is annotated as (*aspect term*, *aspect*, *negative*). **E2** is an example of this category in which the common aspect of question and answer texts is *battery* that is described as "*endurance*" and the answer text expresses negative sentiment to it.

E2: Q: How do you feel about its running? What about the endurance of this phone?

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

A: Yes, the battery is not durable, and its memory is not as large as it is said.

(c) If there exists sentimental expressions like "*good*" and "*great*" in the answer text and they are related to specific aspect in the question text, the QA text pair is annotated as (*aspect term*, *aspect*, *positive*). **E3** is an example of this category. The aspect terms "*certified product*" and "*choppy*" with respect to aspects *certified product* and *performance* respectively are not fully referred in the answer text, which only positively responds to the question about *performance*.

E3: Q: Is it a certified product? Is your phone choppy when you rotate it?
A: My phone is not stuttering, but its screen is not very user-friendly.

(d) Given an aspect in the question text, if the answer text contains both positive and negative sentiment, the QA text pair is annotated as (*aspect term*, *aspect*, *conflict*). **E4** is an example of this category which expresses both positive and negative sentiment about the aspect term "*screen*" relating to the aspect *IO*.

E4: Q: How about this phone? How about its screen?
A: Its screen resolution is very good but its screen is a little bit small, and the battery is not durable.

We assign two annotators to label each QA text pair, and the Kappa consistency check value of the annotation is 0.81. To deal with the QA text pairs which are inconsistent annotated by two annotators, an expert is assigned to check them to ensure the quality of data annotation. After annotation of our corpus, we obtain 2,566 QA text pairs which conform to the above guidelines. In this paper, we mainly focus on identifying the aspect category of a given QA text pair. And we will release this annotated corpus if this paper is accepted.

4 Hierarchical Attention Network

In this section, we firstly describe how to segment both question and answer texts into sentences of each QA text pair. Then, we propose a hierarchical attention network to identify the aspect category of a given QA text pair. Figure 2 depicts the overall architecture of our proposed approach to solving the QA aspect classification task.

As described in Section 3, in a QA text pair, sentences inside question and answer text could contain different aspects. Thus, we use Stanford CoreNLP toolkit [5] to perform sentence segmentation. After sentence segmentation, we then match the sentence inside question text with the sentence inside answer text to construct (sentence, sentence) units.

The core of the proposed approach lies in the hierarchical attention network, which consists of two layers, i.e., QA matching attention layer and self-matching attention layer (see Figure 3). We will describe the two layers respectively in the following sections.

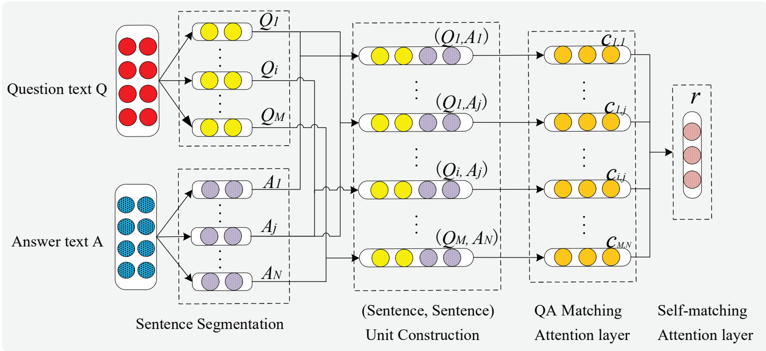


Fig. 2. The overview of our approach.

4.1 QA Matching Attention

Figure 3(a) shows the details of QA matching attention layer. In a QA text pair, assume that the question text Q has been segmented into M sentences $\{Q_1, \cdots, Q_M\}$ and each sentence contains U words. The vector representation $q_{ij} \in R^{d_w}$ denotes the j-th word of the i-th question sentence. And the answer text has been segmented into N sentences $\{A_1, \cdots, A_N\}$ and each sentence contains V words. The vector representation $a_{ij} \in R^{d_w}$ denotes the j-th word of the i-th answer sentence, where d_w represents the dimension of word embeddings in question/answer text.

Then, for $(Q_i,\ A_j)$ unit, we use LSTM [12] model to encode the question sentence Q_i and the answer sentence A_j, where $i \in [1, M]$, and $j \in [1, N]$, and obtain the hidden state matrix $H_{Q_i} = [h_{i1}, \cdots, h_{iU}]$ of Q_i and the hidden state matrix $H_{A_j} = [h_{j1}, \cdots, h_{jV}]$ of A_j, i.e.,

$$H_{Q_i} = \mathrm{LSTM}(Q_i) \tag{1}$$

$$H_{A_j} = \mathrm{LSTM}(A_j) \tag{2}$$

where $H_{Q_i} \in R^{N_w \times d_h}$, $H_{A_j} \in R^{N_w \times d_h}$, N_w is the number of words in a sentence inside question/answer text and d_h is the size of hidden layers.

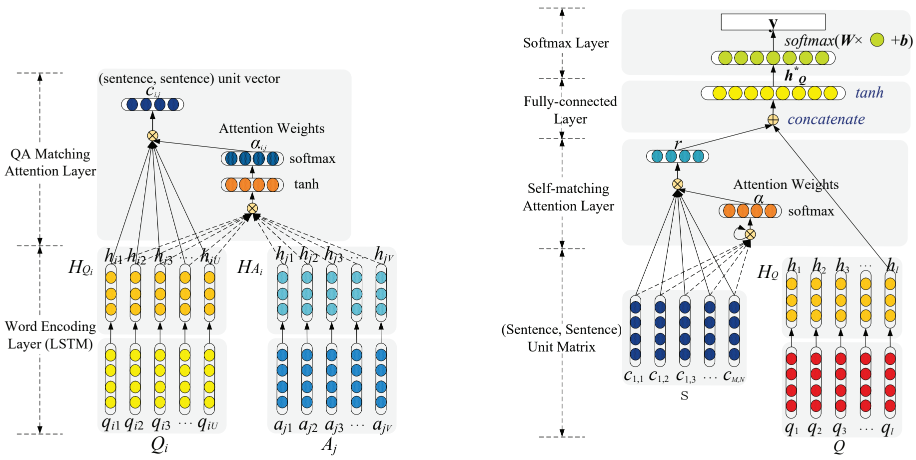
Further, the following formulas are applied to compute the attention weight vector α_{ij} between H_{Q_i} and H_{A_j} to concentrate on such words in question sentence Q_i and answer sentence A_j with respect to the annotated aspect and capture the matching information between them.

$$M_{i,j} = \tanh(W_{ij} \cdot ({H_{Q_i}}^T \cdot H_{A_j}) + b_{ij}) \tag{3}$$

$$\alpha_{i,j} = \mathrm{softmax}({W_e}^T \cdot M_{i,j}) \tag{4}$$

where $M_{i,j} \in R^{N_w \times N_w}$, $\alpha_{ij} \in R^{N_w}$. W_{ij} and W_e are the weight matrices, b_{ij} is the bias and $\cdot$ denotes the dot product between matrices.

From **Guideline1** and **Guideline2** in Section 3, the question text is important for QA aspect classification. Thus, we obtain the (sentence, sentence)



(a) QA matching attention. (b) Self-matching attention.

Fig. 3. The overall architecture of hierarchical attention model.

unit vector $c_{i,j} \in R^{d_h}$ of $(Q_i,\ A_j)$ unit with the hidden state matrix H_{Q_i} of the question sentence Q_i based on the weights, i.e.,

$$c_{i,j} = H_{Q_i} \cdot \alpha_{i,j}^T \tag{5}$$

Finally, we obtain (sentence, sentence) unit vector set $C = \{c_{1,1}, \cdots, c_{i,j}, \cdots, c_{M,N}\}$, where $|C|$ is $M * N$.

4.2 Self-matching Attention

In this layer, we introduce another attention mechanism to automatically determine the importance of each $(Q_i,\ A_j)$ unit, where $i \in [1, M]$, and $j \in [1, N]$, as shown in Figure 3(b).

First, we concatenate multiple (sentence, sentence) unit vectors $\{c_{1,1}, \cdots, c_{i,j}, \cdots, c_{M,N}\}$ together into a new (sentence, sentence) unit matrix $s \in R^{N_c \times d_h}$, where N_c is $M * N$, i.e.,

$$s = [c_{1,1}, \cdots, c_{M,N}] \tag{6}$$

Further, we perform self-matching to obtain the importance degree vector $\alpha \in R^{N_c}$ of different (sentence,sentence) units by the following formulas,

$$M = \tanh(W_s \cdot (s^T \cdot s) + b_s) \tag{7}$$

$$\alpha = \text{softmax}({W_f}^T \cdot M) \tag{8}$$

where $M \in R^{N_c \times N_c}$, W_s and W_f are the weight matrices, b_s is the bias and $\cdot$ denotes the dot product between matrices.

After obtaining self-matching attention weights, we can get the attention representation $r \in R^{d_h}$ of (sentence, sentence) units as follows,

$$r = s \cdot \alpha^T \tag{9}$$

Besides, from **Guideline1** and **Guideline2** in Section 3, the aspect mainly exists in the question text. Therefore, we use LSTM model to encode the original question text Q that is not split and get its hidden state $H_Q = [h_1, \cdots, h_l]$. And we then concatenate the attention representation r of (sentence, sentence) units with the last hidden vector h_l of Q to generate the final representation h_Q^* of Q as follows,

$$h_Q^* = \tanh(W_p r + W_x h_l) \tag{10}$$

where $H_Q \in R^{N_q \times d_h}$, and N_q represents the number of words in the question text. $h_Q^* \in R^{d_h}$, W_p and W_x are the weight matrices.

Finally, we put the final representation h_Q^* into a *softmax* layer to obtain the conditional probability distribution:

$$y = \mathrm{softmax}(W h_Q^* + b) \tag{11}$$

where W and b are parameters for the *softmax* layer. Based on that, the label with the highest probability stands for the predicted aspect of a QA text pair.

4.3 Model Training

Cross-entropy loss function is used to train our model end-to-end for classification. Given a set of training data S_{Q_t}, S_{A_t} and y_t, where S_{Q_t} is the t-th question text, S_{A_t} is the corresponding answer text, and y_t is the ground-truth aspect for a QA text pair (S_{Q_t}, S_{A_t}), if we represent this model as a black-box function $\phi(S_Q, S_A)$ whose output is a vector representing the probability of aspects, then the goal of training is to minimize the loss function:

$$J(\theta) = -\sum_{t=1}^{N_s}\sum_{k=1}^{K} y_t^k \cdot \log \phi(S_{Q_t}, S_{A_t})) + \frac{l}{2}||\theta||_2^2 \tag{12}$$

where N_s is the number of training samples, K is the number of aspects for classification and l is a L_2 regularization to bias parameters.

In the equation above, parameters in our model are optimized by *Adagrad* optimizer [1]. All the matrix and vector parameters are initialized with uniform distribution $[-\sqrt{6/(r+c')}, \sqrt{6/(r+c')}]$, where r and $c^{'}$ are rows and cols of the matrix respectively [2]. Besides, the dropout strategy is used in LSTM layer to avoid over-fitting.

5 Experimentation

5.1 Experimental Settings

- **Data Settings:** Considering the imbalanced distribution of data, we omit the aspect categories which contain less than 50 QA text pairs. In our experiments, there are 7 aspects and 2,427 QA text pairs in total, as shown in

Table 1. The distribution of experimental data.

Aspect Category	Number of QA text pairs
Performance	548
Battery	230
IO	908
Function	111
Quality	165
Certified product	370
Computation	95

Table 1. We also set aside 10% from the training data as the development data to tune learning algorithm parameters.

- **Word Representation:** *Wording Embedding* is used to initialize the words of our dataset and we use skip-gram model to pre-train the word embeddings using 320 thousand QA text pairs crawled from "*Asking All*" in Taobao.
- **Evaluation Metrics:** The performance is evaluated by using *Accuracy* and *Macro-F1* (F) which is calculated by the formula $F = \frac{2PR}{P+R}$, where the overall precision P and recall R are averaged on the precision/recall scores of all categories. Furthermore, *t*-test is used to evaluate the significance of the performance difference between two approaches [15].
- **Hyper-parameters:** In our experiment, all out-of-vocabulary words are initialized by sampling from the uniform distribution U(-0.01,0.01). The dimensions of word embedding and LSTM hidden states are set to be 100. The other hyper-parameters are tuned according to the development data. Specifically, the learning rate is 0.02. The dropout rate is set to 0.25.

5.2 Experimental Results

For a thorough comparison, we implement the following baseline approaches to aspect classification to evaluate the performance of our proposed approach. Note that all these approaches employ the same representations.

- **CNN(A):** This approach is a basic baseline approach proposed by Kim et al. [4] which takes answer texts as input of CNN.
- **CNN(Q):** This approach is a basic baseline approach which takes question texts as input of CNN.
- **CNN(Q+A):** This approach is a basic baseline approach which takes the concatenation of question and answer texts as the input of CNN.
- **LSTM(A):** This baseline approach puts answer texts into the input layer of LSTM proposed by Tang et al. [10].
- **LSTM(Q):** This baseline approach puts question texts into the input layer of LSTM.
- **LSTM(Q+A):** This baseline approach puts the concatenation of question and answer texts into the input layer of LSTM.

Table 2. *Accuracy* and *Macro-F1* on QA aspect classification.

Approaches	*Accuracy*	*Macro-F1*
CNN(A)(Kim et al. [4])	0.575	0.294
CNN(Q)	0.744	0.585
CNN(Q+A)	0.771	0.595
LSTM(A)(Tang et al. [10])	0.663	0.466
LSTM(Q)	0.779	0.668
LSTM(Q+A)	0.800	0.678
Hierarchical LSTM(Xia et al. [13])	0.825	0.733
LSTM-Attention(Wang et al. [12])	0.827	0.729
QA Matching Attention(ours)	0.842	0.751
Hierarchical Attention(ours)	**0.873**	**0.804**

- **Hierarchical LSTM:** This is a question classification approach proposed by Xia et al. [13], which uses a hierarchical LSTM model to encode the question texts.
- **LSTM-Attention:** This baseline approach introduces attention mechanism to capture the aspect matching information between question texts and answer texts proposed by Wang et al. [12].
- **QA Matching Attention:** This is our proposed approach which only employs a QA matching attention layer without a self-matching attention layer.
- **Hierarchical Attention:** This is our proposed approach which employs both QA matching attention layer and self-matching attention layer.

Table 2 gives the performance comparison of different approaches. From Table 2, we can find that:

First, according to the first six approaches, all approaches training using question texts perform better than those training using answer texts, which indicates that question texts contain richer aspect information than answer texts and this conforms to annotation guidelines **Guideline1** and **Guideline2** in Section 3.

Second, when using the concatenation of question and answer texts as input, both **CNN** and **LSTM** methods could obtain better performance, which indicates that answer texts can assist question texts to further improve the performance of QA aspect classification.

Third, we can find that approaches based on **LSTM** perform better than those based on **CNN**. Therefore, LSTM is more suitable for QA aspect classification task than CNN.

Therefore, in the last four approaches, LSTM model is used to encode in regardless of question or answer texts. Moreover, in our proposed approaches **QA Matching Attention** and **Hierarchical Attention**, we take both question and answer texts as input. We could find that the performance of **Hierarchical LSTM** method is better than **LSTM** but worse than **LSTM-Attention** method, and the accuracy of **LSTM-Attention** method is 4.8% higher than that of **LSTM** method, which indicate that attention mechanism is a good choice to capture the information related to the specific aspect.

Besides, our proposed **QA Matching Attention** approach achieves the improvement of 1.5% (Accuracy) and 2.2%(*Macro-F1*) compared to **LSTM-Attention** method, which highlights the importance of employing the aspect matching information between the sentence inside question text and the sentence inside answer text.

Noted that, our proposed **Hierarchical Attention** approach using both QA Matching Attention layer and self-matching attention layer achieves the best performance than all other approaches. The accuracy and *Macro-F1* of **Hierarchical Attention** approach are 3.1% and 5.3% higher than **QA Matching Attention** only using QA Matching Attention layer respectively, which indicates that using self-matching attention layer can effectively capture the different importance degrees of different (sentence, sentence) units in a QA text pair and thus could further boost the performance of QA aspect classification. Significance test shows that this improvement is significant ($p - value < 0.05$).

5.3 Error Analysis

Based on the analysis of misclassified QA text pairs, we can find some main reasons for misclassification as follows:

(1) The distribution of experimental data is imbalanced (see Table 1), so the most of misclassified QA text pairs tend to be predicted as the aspect categories accounting for a large rate in experimental data. According to statistics, in our proposed approach, the predicted aspect of 22.95% of misclassified QA text pairs is *IO* and 16.39% is *performance*.

(2) Some manually annotated aspect terms are ambiguous. We can factitiously determine their different meanings and categorize them into correct categories, while it is difficult for the well-trained machine. For instance, the aspect term *sound quality* is related to the aspect *IO*, but the machine may regard it relating to the aspect category *quality*.

(3) The word segmentation toolkit cannot accurately segment out-of-vocabulary words, such as *mhl* and *wifi*, which may also affect the performance of QA aspect classification.

6 Conclusion

In this paper, we propose a hierarchical attention network for aspect classification of QA text pairs. The main idea of the proposed approach is to segment question and answer texts into sentences and then extend feature representations of question texts with the two layers of QA matching attention and self-matching attention. Experimental results demonstrate that our proposed approach outperforms some strong baseline methods using neural networks.

In our future work, we would like to perform joint learning on both aspect classification and aspect term extraction tasks to further improve the performance of QA aspect classification.

Acknowledgements

This work is supported in part by Industrial Prospective Project of Jiangsu Technology Department under Grant No.BE2017081 and the National Natural Science Foundation of China under Grant No.61572129.

References

1. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(Jul), 2121–2159 (2011)
2. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the 13th international conference on artificial intelligence and statistics. pp. 249–256 (2010)
3. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: An unsupervised neural attention model for aspect extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. vol. 1, pp. 388–397 (2017)
4. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on EMNLP. pp. 1746–1751 (2014)
5. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd Annual Meeting of ACL: System Demonstrations. pp. 55–60. ACL (2014)
6. Mukherjee, A., Liu, B.: Aspect extraction through semi-supervised modeling. In: Proceedings of the 50th Annual Meeting of ACL. pp. 339–348. ACL (2012)
7. Poria, S., Cambria, E., Ku, L., Gui, C., Gelbukh, A.: A rule-based approach to aspect extraction from product reviews. In: Proceedings of the Second Workshop on Natural Language Processing for Social Media. pp. 28–37 (2014)
8. Rana, T.A., Cheah, Y.: A two-fold rule-based model for aspect extraction. Expert Syst. Appl. **89**, 273–285 (2017)
9. Shu, L., Xu, H., Liu, B.: Lifelong learning crf for supervised aspect extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. vol. 2, pp. 148–154 (2017)
10. Tang, D., Qin, B., Feng, X., Liu, T.: Effective lstms for target-dependent sentiment classification. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 3298–3307 (2016)
11. Toh, Z., Su, J.: Nlangp: Supervised machine learning system for aspect category classification and opinion target extraction. In: International Workshop on Semantic Evaluation. pp. 496–501 (2015)
12. Wang, Y., Huang, M., Zhao, L., et al.: Attention-based lstm for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 606–615 (2016)
13. Xia, W., Zhu, W., Liao, B., Chen, M., Cai, L., Huang, L.: Novel architecture for long short-term memory used in question classification. Neurocomputing (2018)
14. Xue, W., Zhou, W., Li, T., Wang, Q.: Mtna: A neural multi-task model for aspect category classification and aspect term extraction on restaurant reviews. In: Proceedings of the 8th International Joint Conference on Natural Language Processing. pp. 151–156. Asian Federation of Natural Language Processing (2017)
15. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 42–49 (1999)