Question-Answering Aspect Classification with Multi-attention Representation

Hanqian Wu[1,2](✉), Mumu Liu[1,2], Jingjing Wang[3], Jue Xie[4], and Shoushan Li[3]

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

Abstract. In e-commerce platforms, the question-answering style reviews are emerging, which usually contains much aspect-related information about products. In this paper, Question-answering (QA) aspect classification is a new task that aims to identify the aspect category of a given QA text pair. According to characteristics of QA-style reviews, we draw up annotation guidelines and build a high-consistency annotated corpus for QA aspect classification. Then, we propose a recurrent neural network based on multi-attention representation to tackle this new task. Specifically, we firstly segment the answer text into clauses, and then leverage the multi-attention representation layer to match the question text with clauses inside answer text and generate multiple attention representations of the question text, which extends feature information of the question text. The experimental results demonstrate that our method for QA aspect classification, which is based on multi-attention representation, can make the most of useful information in answer texts and perform better than some strong baselines in QA aspect classification.

Keywords: Question answering · Aspect classification
Attention mechanism

1 Introduction

Recently, there appears a large number of user-generated question-answering (QA) reviews in various e-commerce platforms, such as Amazon, Taobao and Kaola. An example of QA-style reviews is shown in Fig. 1. By this novel form of reviewing, the potential customers can ask questions about certain product and others who have purchased the same item kindly answer these questions. Thus, this QA-style reviews are more reliable and convincing than traditional reviews written by any users. However, very few research has been conducted on aspect

S. Zhang et al. (Eds.): CCIR 2018, LNCS 11168, pp. 78–89, 2018.
https://doi.org/10.1007/978-3-030-01012-6_7

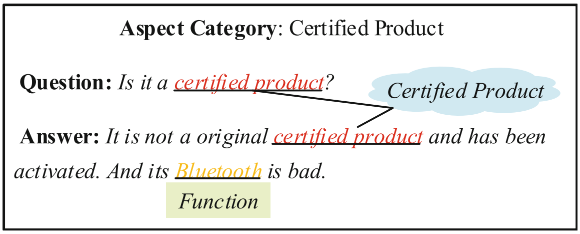

Fig. 1. A translated example of QA-style reviews from an e-commerce platform.

classification of QA-style reviews which aims to identify the aspect category of a given QA text pair.

According to analysis of corpus, as shown in Fig. 1, we can find that **Question** only contains one aspect *certified product*, while there are two different aspects contained in different clauses in **Answer**, i.e., the clause "*It is not a original certified product and has been activated.*" relating to the aspect *certified product* and the clause "*And its Bluetooth is bad*" relating to the aspect *function*. From the point view of consumers questioning, the expected answer should be only related to the aspect *certified product*. Thus, for this QA text pair, we should only consider how to identify the aspect *certified product* involved in **Answer**. Inspired by this, we can firstly segment *Answer* into clauses to make each clause contain only one aspect. Then we can leverage a attention-based layer to match **Question** with each clause in **Answer** and capture the most relevant information between **Question** and **Answer**.

In view of the above issue, we first draw up some specifically designed annotation guidelines and build a high-consistency corpus for our QA aspect classification task. On this basis, we propose a recurrent neural network based on multi-attention representation to solve this task. Specifically, we segment the answer text into different clauses, and then leverage the multi-attention representation layer to match the question text with each clause inside answer text and obtain multiple attention representations of the question text to extend the features of question text for classification. The empirical studies demonstrate that our proposed approach can make full use of the relevant information in the answer text to achieve better performance and outperform other baseline methods.

2 Related Work

Aspect classification, i.e., aspect category classification, can be treated with approaches applied to text classification, such as CNN [5], LSTM [11]. However, there is very few research with focus on aspect classification task. Toh et al. [12] leverage the sigmoidal feedforward network to train binary classifiers for aspect category classification. Xue et al. [15] perform joint learning with the two tasks aspect category classification and aspect term extraction based on neural networks.

In addition, most researchers focus on extracting the more fine-grained opinion targets from reviews, namely aspect extract task. This task is associated with aspect classification task, which also aims to identify the aspect category of a given review text. The difference is that aspect extraction task usually includes two subtasks, i.e., extracting all aspect terms from corpus and clustering aspect terms with similar meaning into aspect categories. Poria et al. [8] propose a rule-based approach that make use of common-sense knowledge and sentence dependency trees to detect both explicit and implicit aspects from opinionated texts. Rana et al. [9] mine sequential patterns from customer reviews and define rules on the basis of these patterns, and they then propose a two-fold rule-based method, in which the first fold extracts aspect associated with domain independent opinions and the second fold extracts aspects associated with domain dependent opinions. Conditional Random Fields (CRF) [6] method requires much manual annotation. Inspired by the approach [1], Shu et al. [10] propose a lifelong CRF model for aspect extraction which leverages the knowledge from many past domains to assist extraction for a new domain. Unsupervised methods are also applied to avoid reliance on annotated data. For instance, in recent years, Latent Dirichlet Allocation (LDA) and its variants have become the dominant unsupervised approach to aspect extraction [7]. However, LDA-based models need to estimate a distribution of topics for each document. To address the above challenge of the LDA-based methods, He et al. [4] propose a attention-based neural approach to emphasize aspect-related words to further improve the coherence of aspects.

Furthermore, as far as we know, our study takes the lead in the aspect classification task for QA text pairs, which distinctly differs from the above existing research.

3 Data Collection and Annotation

Our data are mainly form "*Asking All*" in Taobao[1], which is the most famous e-commerce platform in China. We extract 8,313 QA text pairs from the *electronic appliances* domain and manually annotate them. To ensure the high consistency of annotation, we draw up three aspect-related annotation guidelines (G1, G2 and G3) and assign two annotators to label each QA text pair in the form of a triple of *aspect term*, *aspect* and *polarity*. Note that all examples presented in this paper are translations of original Chinese texts.

G1: If we extract the aspect term from the question text, we will consider to annotate this QA text pair with a triple. According to the headword of an aspect term, the extraction fineness of aspect terms can be divided into the following situations.

(a) If the headword is a verb, the extraction fineness of the aspect term is the verb with a noun which follows closely.

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

(b) If the headword is a noun, the extraction of the aspect term will conform to the principle of noun phrase maximization.
(c) If the headword is an adjective or adverb, the aspect term is the adjective or adverb.

G2: All aspects, i.e., aspect categories, are predefined based on the extraction of aspect terms in our corpus. For instance, aspect terms "*screen*" and "*signal*" extracted from question texts can be classified as the aspect category "*IO*". Aspects are divided into two categories, one is domain-independent aspect, i.e., the aspect may appear in 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* which only exist in *electronic appliances* domain.

G3: Once an aspect is mentioned in both the question and answer text, the next step is to decide the sentiment polarity of this aspect and then annotate it with a triple. Generally speaking, the sentiment polarity can be subdivided into *positive*, *negative* and *conflict* (a mix of both positive and negative) and *neutral* categories. We may come across the following cases during the processing of annotation,

(a) If the answer text expresses the objective evaluation about the aspect referred 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, the aspect term is related to the aspect *appearance*, and the clause "*Its back is not very flat,*" objectively evaluates it, so we annotate this QA text pair with a triple of (*flat*, *appearance*, *neutral*).
  E1: Q: Is the back of this phone flat?
  A: Its back is not very flat, but it feels good. And its signal is too bad.
(b) If the answer text contains negative sentimental words related to the aspect in the question text, such as "*too bad*" and "*not good*", the QA text pair is annotated as (*aspect term*, *aspect*, *negative*). **E2** is an example of this category. The aspect term "*choppy*" is related to the aspect *performance* involved in both the question and answer text, and the answer text expresses negative sentiment of it. Though the answer text also expresses the negative sentiment of camera pixels, the question text does not refer to pixels. Thus, this QA text pair is annotated as (*choppy*, *performance*, *negative*).
  E2: Q: Is it choppy when you are playing games?
  A: It fails to work well and is choppy. And the pixels of camera are low.
(c) If there exists sentimental expressions like "*great*" and "*good*" 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.
  E3: Q: Does this phone peel off paint easily?
  A: No, it does not and I have been using it for a long time. And its signal is good.

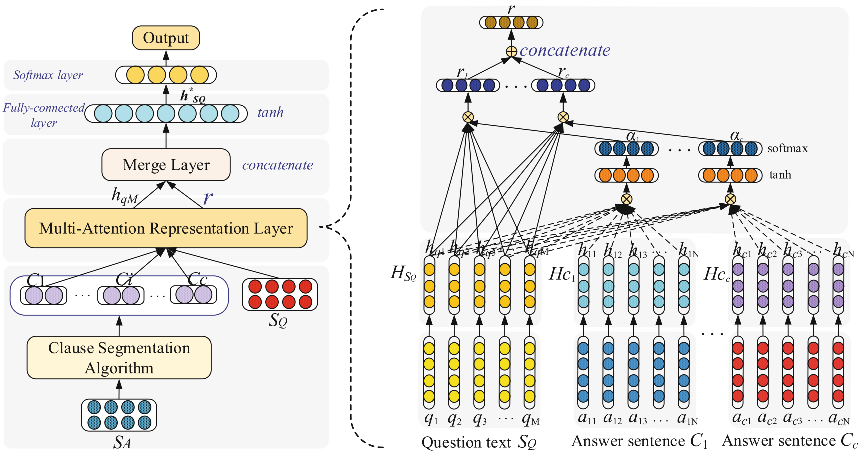

Fig. 2. The overall architecture of our approach.

(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 "*signal*" relating to the aspect *IO*.

E4: Q: How about its signal?
A: It has a good semaphore when using the mobile phone, but the connection of wireless network fails to work well. And it makes its memory card easily hot.

The Kappa consistency check value of the annotation is 0.81. To cope with the inconsistently annotated QA text pairs, an expert is assigned to proofread them. After annotation, we finally build a high-consistency corpus with annotated 2,566 QA text pairs which conform to the above annotation guidelines. In this paper, our goal is to identify the aspect category of a given QA text pair. And if this paper is accepted, we will release this high-consistency annotated corpus.

4 Our Approach

In this section, we firstly introduce the clause segmentation algorithm of the answer text, and then we propose a recurrent neural network based on multi-attention representation to identify the aspect category of a given QA text pair. Figure 2 shows the overall architecture of our proposed approach to the QA aspect classification task. We will describe our approach in detail in the following sections.

Algorithm 1: Clause Segmentation Algorithm

Input: Answer text S_A={w_i | w_i is a word.};
V_p: Chinese punctuations set;
N_{min}: the minimum number of words in a clause;
N_{max}: the maximum number of clauses in the answer text

Output: All clauses (Stored in C={ c_i }) mined from S_A that satisfy N_{min} and N_{max}.

$C = \emptyset$;
c_{temp}=*null*; //the candidate clause
Segment S_A into n clauses $\{a_1, \cdots, a_n\}$ with V_p;
for $i = 1$; $i \leq |S_A| - 1$; $i+ = 1$ **do**
    if $|C| \geq N_{\max}$ **then**
        break;
    end
    if $a_i.length > N_{\min}$ **then**
        $c_{temp} = a_i$;
        C=$C \cup \{c_{temp}\}$;
    else
        $j = i + 1$;
        while $j \leq |S_A| - 1$ **do**
            $c_{temp} = a_i + a_j$;
            if $c_{temp}.length \geq N_{\min}$ **then**
                C=$C \cup \{c_{temp}\}$;
            else
                j+=1;
            end
        end
        $i = i + j$;
    end
end

4.1 Clause Segmentation Algorithm

As described in Sect. 3, clauses inside answer text could contain different aspects in a QA text pair and only one clause is related to the annotated aspect. Thus, we segment the answer text into clauses to capture useful information contained in the answer text for classification.

The main idea of clause segmentation algorithm is to segment answer texts with Chinese punctuations as delimiters. For a clause in the answer text, we define the minimum number of words as N_{min}. And only when the length of one clause is larger than N_{min}, it is called a clause. Besides, we define the maximum number of clauses as N_{max} to determine the number of clauses required for the input of the neural network. Algorithm 1 describes the clause segmentation algorithm in detail.

4.2 Multi-attention Representation Layer

The core of our proposed approach is to capture the most relevant information between the question and answer text by leveraging the multi-attention representation layer and extend the feature representation of the question text to improve the performance of QA aspect classification. The right part in Fig. 2 depicts how to obtain the multiple attention representations of the question text in details.

For a given QA text pair, assume that the answer text S_A has been segmented into c clauses $\{c_1, \cdots, c_c\}$ and each clause contains N words. The vector representation $a_{ij} \in R^{d_w}$ denotes the j-th word of the i-th answer clause. The question text S_Q contains M words, and the vector representation $q_i \in R^{d_w}$ denotes the i-th word in the question text, where d_w represents the dimension of word embeddings in the question/answer text.

First, we encode the question text S_Q and the answer clause c_i with LSTM model [11,13], where $i \in [1, c]$, to obtain the hidden state matrix $H_{S_Q} = [h_{q1}, \cdots, h_{qM}]$ of S_Q and $H_{c_i} = [h_{i1}, \cdots, h_{iN}]$ of c_i by the following formulas,

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

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

where $H_{S_Q} \in R^{N_w \times d_h}$, $H_{c_i} \in R^{N_w \times d_h}$, N_w is the number of words in the question text or the answer clause, and d_h is the size of LSTM hidden layer.

Further, we compute the attention weight vector α_i between H_{S_Q} and H_{c_i} to capture the most relevant information relating to the annotated aspect between question sentence S_Q and answer clause c_i as follows,

$$M_i = \tanh(W_i \cdot ({H_{S_Q}}^T \cdot H_{c_i}) + b_i) \tag{3}$$

$$\alpha_i = \mathrm{softmax}({W_e}^T \cdot M_i) \tag{4}$$

where $1 \le i \le C$, $M_i \in R^{N_w \times N_w}$, $\alpha_i \in R^{N_w}$, W_i and W_e are the weight matrices, b_i is the bias and $\cdot$ denotes the dot product between matrices.

Then, we obtain the attention representation $r_i \in R^{d_h}$ of the question text S_Q based on the weights, i.e.,

$$r_i = H_{S_Q} \cdot \alpha_i^T \tag{5}$$

The answer text is segmented into c clauses, so we can obtain the attention representation set $R = \{r_1, \cdots, r_i, \cdots, r_c\}$ of the question text S_Q where $|R|$ is c. And we concatenate these attention representations together into a new vector $r \in R^{d_h}$.

$$r = r_1 \oplus \cdots \oplus r_i \oplus \cdots \oplus r_c \tag{6}$$

Besides, according to the guidelines **G1** and **G2** in Sect. 3, QA aspect classification task mainly depends on the question text. Thus, the final feature representation $h^* \in R^{d_h}$ of S_Q is computed by concatenating r with the last hidden vector $h_{qM} \in R^{d_h}$ of the question text as follows,

$$h^* = \tanh(W_p r + W_x h_{qM}) \tag{7}$$

where W_p and W_x are the weight matrices.

Finally, a *softmax* layer is followed to obtain the conditional probability distribution:

$$y = \text{softmax}(W h^* + b) \tag{8}$$

where W and b are parameters for the *softmax* layer. On this basis, the label with the highest probability stands for the predicted aspect category 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 optimization goal of training is to minimize the loss function:

$$J(\theta) = -\sum_{t=1}^{N_s}\sum_{k=1}^{K} y_t^k \cdot \log \phi(S_{Q_t}, S_{A_t})) + \frac{l}{2}||\theta||_2^2 \tag{9}$$

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, we adopt *Adagrad* optimizer [2] to optimize parameters in our model, and initialize all the matrix and vector parameters with uniform distribution $[-\sqrt{6/(r+c')}, \sqrt{6/(r+c')}]$, where r and $c^{'}$ are rows and cols of the matrix respectively [3]. Besides, the dropout strategy is used in LSTM layer to avoid over-fitting.

5 Experimentation

5.1 Experimental Settings

- **Data Settings:** Due to the imbalance of distribution of data, aspect categories which contain less than 50 QA text pairs are omitted. Table 1 depicts the distribution of experimental data. Besides, we set aside 10% from the training data as the development data to tune learning algorithm parameters.
- **Word Representations:** Word embedding is used for feature representations of experimental data, which is pre-trained based on Skip-Gram model with Gensim [3] toolkit and 320 thousand QA text pairs collected from "*Asking All*" in Taobao.

Table 1. Data distribution in our experiment.

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

- **Evaluation Metrics:** The evaluation metrics of performance are mainly *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 the average of the precision/recall scores of all categories. Furthermore, we use t-test to assess the significance of the performance difference between two approaches [16].
- **Hyper-parameters:** In our experiment, the dimensions of word embeddings and LSTM hidden layers are set to be 100. The other hyper-parameters are tuned according to the development data. Specifically, the learning rate is 0.01 and the dropout rate is 0.4. And in our clause segmentation algorithm, the minimum number of words in a clause is 5 and the maximum number of clauses inside the answer text is 3. Besides, all out-of-vocabulary words are initialized by sampling from the uniform distribution $U(-0.01, 0.01)$.

5.2 Experimental Results

For a comprehensive analysis and comparison, we implement some baselines for QA aspect classification task to evaluate the performance of our proposed approach. And all approaches use the same word representations.

- **CNN(A):** This basic baseline approach proposed by Kim et al. [5] takes answer texts as input of CNN.
- **CNN(Q):** This basic baseline approach takes question texts as input of CNN.
- **CNN(Q+A):** This basic baseline approach takes the concatenation of question and answer texts as the input of CNN.
- **LSTM(A):** This is a baseline approach which puts answer texts into the input layer of LSTM proposed by Tang et al. [11].
- **LSTM(Q):** This is a baseline approach which puts question texts into the input layer of LSTM.
- **LSTM(Q+A):** This is a baseline approach which puts the concatenation of question and answer texts into the input layer of LSTM.
- **Hierarchical LSTM:** This baseline approach is used for question classification proposed by Xia et al. [14], which uses a hierarchical LSTM model to encode the question texts for classification.

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

Approaches	*Accuracy*	*Macro-F1*
CNN(A) (Kim et al. [5])	0.575	0.294
CNN(Q)	0.744	0.585
CNN(Q+A)	0.771	0.595
LSTM(A) (Tang et al. [11])	0.675	0.468
LSTM(Q)	0.804	0.665
LSTM(Q+A)	0.850	0.706
Hierarchical LSTM (Xia [14])	0.827	0.729
Individual-Attention (Wang et al. [13])	0.835	0.755
Multi-Attention (ours)	0.865	0.818

- **Individual-Attention:** This baseline approach leverages attention mechanism to capture the relevant information between the question and answer text without clause segmentation proposed by Wang et al. [13].
- **Multi-Attention:** This is our proposed approach which introduces multi-attention mechanism with clause segmentation.

Table 2 demonstrates the experimental results of all approaches in our experiment. By analysis, we can draw some conclusions as follows:

First, by analyzing the approaches based on **CNN** and **LSTM**, we can find that approaches only using question texts as input all outperform those only using answer texts for classification, which accords with annotation guidelines **G1** and **G2** in Sect. 3.

Second, in the approaches **CNN** and **LSTM**, approaches with the concatenation of question and answer texts as input are better than other methods, which demonstrates that answer texts can assist question texts and bring performance improvement for QA aspect classification.

Third, we can find that the performance of approaches based on **LSTM** is obviously superior to that of approaches based on **CNN**. Thus, LSTM is better for QA aspect classification task than CNN.

Therefore, the last three approaches are based on LSTM and take question texts as input. And the two approaches **Individual-Attention** and **Multi-Attention** utilize the relevant information in answer texts based on attention mechanism. The approach **Hierarchical LSTM** performs better than **LSTM** but worse than **Individual-Attention**. And the **Individual-Attention** approach achieves the improvement of 3.1% (Accuracy) and 9.0% (*Macro-F1*) compared with the **LSTM** approach, which proves that it is a good choice to introduce attention mechanism to capture relevant information with respect to the annotated aspect between the question and answer text.

Our proposed **Multi-Attention** approach outperforms most of all approaches, and the accuracy and *Macro-F1* of our model are respectively 3% and 6.3% higher than those of the method **Individual-Attention**. The empirical studies demonstrate that our proposed approach in which we introduce multiple attention representations based on clause segmentation algorithm, can capture the most aspect-related information between the question and answer texts in a QA text pair so as to achieve better performance for QA aspect classification. Significance test shows that this improvement is significant ($p - value < 0.05$).

5.3 Parameter Analysis

The hyper-parameters tuned according to the development data are optimal and different parameters may affect the performance of our proposed approach. The key parameter in our experiment is the number of clauses in the answer text N_{max}. When the other parameters are fixed, we tune the parameter N_{max} and find that the best value of N_{max} is 3. Figure 3 depicts the line chart of *Accuracy* and *Macro-F* changing with N_{max}.

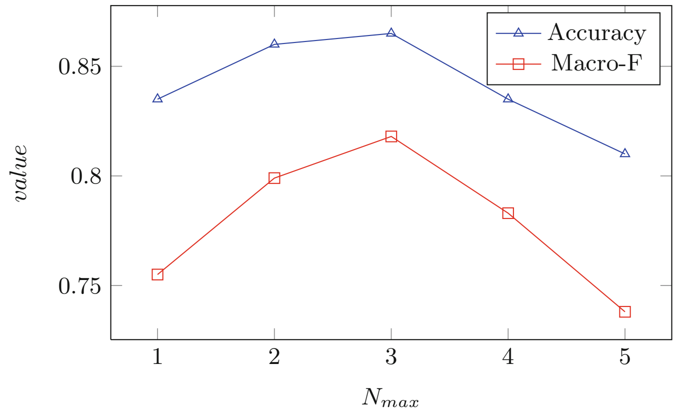

Fig. 3. The line chart of *Accuracy* and *Macro-F* changing with N_{max}.

6 Conclusion

QA aspect classification task is essential to QA aspect-level sentiment analysis. The characteristic of our proposed approach is that we introduce multi-attention mechanism based on clause segmentation to generate multiple attention representations of the question text, which extends the feature representation of the question text to further improve the performance of QA aspect classification. The experimental results demonstrate that the Multi-Attention representation method outperforms some strong baseline methods neural networks.

In our future work, we will consider to perform joint learning with the two tasks QA aspect classification and aspect term extraction to further achieve better performance for 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. Chen, Z., Liu, B.: Topic modeling using topics from many domains, lifelong learning and big data. In: International Conference on Machine Learning, pp. 703–711 (2014)
2. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**(Jul), 2121–2159 (2011)
3. 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)
4. 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)
5. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on EMNLP, pp. 1746–1751 (2014)
6. Mitchell, M., Aguilar, J., Wilson, T., Van Durme, B.: Open domain targeted sentiment. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1643–1654 (2013)
7. Mukherjee, A., Liu, B.: Aspect extraction through semi-supervised modeling. In: Proceedings of the 50th Annual Meeting of ACL, pp. 339–348. ACL (2012)
8. 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)
9. Rana, T.A., Cheah, Y.: A two-fold rule-based model for aspect extraction. Expert Syst. Appl. **89**, 273–285 (2017)
10. 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)
11. 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)
12. 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)
13. 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)
14. Xia, W., Zhu, W., Liao, B., Chen, M., Cai, L., Huang, L.: Novel architecture for long short-term memory used in question classification. Neurocomputing **299**, 20–31 (2018)
15. 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)
16. 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)