Cluster Computing (2020) 23:1973–1986
https://doi.org/10.1007/s10586-020-03160-9

Multi-task learning based on question–answering style reviews for aspect category classification and aspect term extraction on GPU clusters

Hanqian Wu[1,2] · Siliang Cheng[1,2] · Zhike Wang[1,2] · Shangbin Zhang[1,2] · Feng Yuan[1,2]

Received: 23 February 2020 / Revised: 16 July 2020 / Accepted: 18 July 2020 / Published online: 27 July 2020

Abstract

Cluster computing technologies are rapidly advancing and user-generated online reviews are booming in the current Internet and e-commerce environment. The latest question–answering (Q&A)-style reviews are novel, abundant and easily digestible product reviews that also contain massive valuable information for customers. In this paper, we mine valuable aspect information of products contained in these reviews on GPU clusters. To achieve this goal, we utilize two subtasks of aspect-based sentiment analysis: aspect term extraction (ATE) and aspect category classification (ACC). Most previous works focused on only one task or solved these two tasks separately, even though they are highly interrelated, and they do not make full use of abundant training resources. To address this problem, we propose a novel multi-task neural learning model to jointly handle these two tasks and explore the performance of our model on GPU clusters. We conducted extensive comparative experiments on an annotated corpus and found that our proposed model outperforms several baseline models in ATE and ACC tasks on GPU clusters, yielding significant strides in data mining for these types of reviews.

Keywords Aspect-based sentiment analysis · Question–answering reviews · Multi-task learning · GPU clusters · Data parallelism

1 Introduction

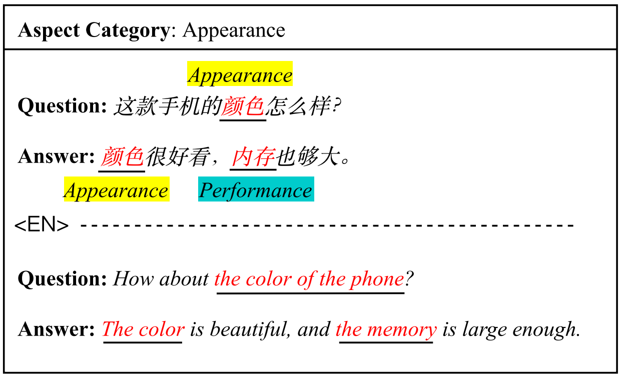
With the advancement of computing technology, GPU cluster as a typical heterogeneous cluster becomes one of the most significant computing infrastructures and is widely applied into scientific computing, information services and big data processing. The advance in hardware contributes a lot of improvements in various fields, including computer vision [6], social media and e-commerce platforms. The evolvement of social media and e-commerce platforms contributes to the popularity of online product reviews exceptionally, thus massive online product reviews are generated. To process and analyze these abundant and textual reviews, Natural Language Processing (NLP) has garnered significant attention in recent years. Aspect-based sentiment analysis, as an important research topic in NLP field, offers fine-grained tasks for mining aspect information of product reviews. Accurately mining this information, however, involves three important subtasks: aspect term extraction (ATE), aspect category classification (ACC), and aspect-level sentiment classification (ASC).

Question–answering (Q&A)-style review, which is a novel form of product review, consists of questions and answers where potential consumers generate questions and sellers or people who purchased the products provide answers. Figure 1 shows an example of Q&A-style reviews with annotation information. Different from conventional reviews which many useless messages might be included in (e.g. "The color is beautiful, the price is low and performance is great. A movie star is endorsing it and you'll regret if you miss it."), Q&A-style reviews are pairs of

✉ Hanqian Wu
hanqian@seu.edu.cn

[1] School of Computer Science and Engineering, Southeast University, Nanjing, China

[2] Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, China



Aspect Category: Appearance

Appearance

Question: *这款手机的颜色怎么样?*

Answer: *颜色很好看，内存也够大。*

Appearance *Performance*

<EN> ---

Question: *How about the color of the phone?*

Answer: *The color is beautiful, and the memory is large enough.*

Fig. 1 An example of a question–answering (Q&A)-style review

conversation, they are more targeted, and the topic of answer texts are confined to the topic of question text. Hence Q&A-style reviews effectively reduce the number of fake reviews and make product information more credible. Thus, aspect-based sentiment analysis is particularly necessary and meaningful for mining valuable information contained in Q&A reviews.

Recently, several studies have focused on ATE and ACC tasks. However, most studies have focused on traditional reviews rather than Q&A-style reviews and regard ATE and ACC as independent tasks and deal with them separately, even though the tasks are highly interrelated. Intuitively, extracted aspect term information assists aspect category prediction, and aspect category information is advantageous to distinguish aspect terms from other words unrelated to aspect information. In addition, to the best of our knowledge, there is no one to explore the ATE task and ACC task based on Q&A-style reviews on GPU clusters. One of the barriers of these two tasks on Q&A-style reviews is the corpus about Q&A-style reviews–especially the Chinese corpus–is scarce. The good news is that we recently resolved this difficulty in previous work by designing a set of elaborate annotation rules and building a high-quality annotated corpus [31, 32]. Beyond that, there are some other important problems needed to be addressed in ATE and ACC tasks.

On the one hand, most studies do not make full use of training resources. Up to now, E-commerce platforms have accumulated massive online product reviews, including Q&A-style reviews. This renders us to get more training sets for our models which can significantly improve the performance of our model. Apart from more training data, we can also utilize more powerful computing power and faster computing speed via GPU clusters. This can greatly improve the quality of the model for it not only enables more data to process during training stage, but also reduces the iteration time in experiment, allows researchers try on their new ideas and configurations. Faster training also enables networks to be deployed in applications whose model needs to be updated frequently. In this paper, we employ data parallelism which will train several mini-batches simultaneously on parallel GPUs. We allocate training data to multiple processors to compute gradient updates, then aggregate these divided updates to get the final result. To the best of our knowledge, we are the first to explore the aspect-based sentiment analysis tasks on GPU clusters.

On the other hand, there are some challenges caused by Q&A-style reviews. First, because of colloquial and informal nature of online reviews, existing word segmentation toolkits generate errors when dealing with text of Q&A-style reviews, which degrades the subsequent model's performance. As a solution, we adopt character-level rather than word-level embedding to represent Q&A text. Second, the ACC task for Q&A-style reviews is more difficult than for conventional reviews, because of the occasional irrelevant aspect terms. With this in mind, it only makes sense to focus on the aspect term mentioned in both the question and answer context. Third, because of the correlation between ATE task and ACC task, extracted aspect term information assists aspect category prediction, and aspect category information is advantageous to distinguish aspect terms from other words unrelated to aspect information. To overcome this problem, we propose a novel multi-task neural learning framework that jointly addresses the two tasks.

In this paper, by analyzing all problems of Q&A reviews mentioned above, our research offers the following contributions:

- To make full use of training resources and improve the training speed of models, we deploy our models and all baselines in GPU clusters, and use data parallel strategies for model training. In this paper, we will compare the performance and training time of our proposed model with other baselines in GPU clusters.
- To contend with colloquial and informal nature of online reviews, we avoid word segmentation errors by adopting character-level rather than word-level embedding. In this way, our proposed model improves the performance of ACC task and implements fine-grained extraction for ATE task.
- To address the occasional irrelevant aspect terms mentioned in the Q&A context, we introduce attention mechanism that captures the most relevant aspect information mentioned by both the question and answer contexts, which improves the performance of ACC task.
- To further improve the performance of ACC and ATE, we leverage the correlation between ATE and ACC to jointly address the two tasks.

2 Related work

2.1 Aspect category classification

The ACC task can be treated as a supervised classification task [20]. Given the predefined categories list, our task is to identify a specified aspect term's category. Traditional approaches mainly focused on manually designing a set of features such as a bag-of-words or lexicon to train a classifier. Brychcin et al. [2] leveraged a set of binary Maximum Entropy (ME) classifiers for ACC. Kiritchenko et al. [13] used a set of binary Support Vector Machines (SVMs) with different types of n-grams and information from a specially designed lexicon. However, these approaches highly depend on the features' quality, and feature engineering is labor-intensive.

With the development of deep learning techniques, researchers have designed effective neural networks to address ACC task. Toh et al. [28] extracted features from words in every sentence and adopted the sigmoidal feed-forward network to train a binary classifier. Xue et al. [34] proposed a multi-task neural network to jointly address ACC and ATE. (Our research differs from the work they built on conventional reviews, our proposed model is based on Q&A-style reviews and we leverage conditional random fields to further improve ATE's performance.) More recently, Wu et al. [33] proposed a 4-dimension textual representation model on Q&A style reviews for ACC task.

2.2 Aspect term extraction

The ATE task extracts aspect and opinion terms explicitly contained in the sentence [25]. Early work focused on researching rule-based methods. Hu and Liu et al. [10] leveraged frequent nouns or noun phrases to extract aspect terms, and tried to identify opinion terms by exploiting the relationships and occurrence between aspect terms and opinion terms. However, the rule-based approaches highly relied on hard-coded rules and external language resources. Later, ATE was treated as sequence tagging problem by using supervised featured-based methods such as Hidden Markov Models (HMMs) [17] or Conditional Random Fields (CRF) [27]. However, feature-based approaches greatly rely on features' quality–and again, feature engineering is both time-consuming and labor-intensive.

With the rapid development of neural networks, researchers proposed a neural language model for general high-level representations of words used to extract aspect terms [9]. Liu et al. [21] used pre-trained word embeddings as input of Recurrent Neural Network (RNN) for ATE. Yin et al. [35] proposed a hybrid method that first learns a distributed representation of words and dependency paths by RNN and then feeds the learned results along with some hand-crafted features into a CRF [16] for extracting aspect terms. Wang et al. [29] proposed a joint model consisting of Recursive Neural Network (ReNN) and CRF layer for ATE task. To reduce the influence of parsing errors, they further designed the RNN with coupled multilayer attention, to exploit the relationship of aspect terms and opinion terms for co-extraction [30]. Recently, Li et al. [19] designed a framework which can exploit opinion summary and aspect detection history for tackling ATE. Ma et al. [22] conducted a gated unit network with attention mechanism to make Seq2Seq learning suit to ATE task. Li et al. [18] alleviated the data scarcity problem in ATE task by proposing a masked sequence-to-sequence data augmentation method.

2.3 Multi-GPU parallel computing

Parallel Computing is a kind of method to solve computing problems by using multiple computing resources simultaneously, which is a useful means to enhance computational efficiency and processing ability of computer system.The history of the utilization of parallel computing can be traced back to 1970s when the first parallel computer ILLIAC IV was invented. By the difference of principle, parallel computing can be divided into Data Parallelism(DP) [3] and Model Parallelism(MP) [4]. In DP, each GPU uses the same model to train on a different subset of training data and compute gradients, which need to be aggregated across the GPUs [15]. The strategy is widely used since its simplicity and highly effectiveness in reducing time cost [5].

Since the growth of machine learning techniques in recent years, researchers have brought parallel computing to the domain. Meyer et al. [23] introduced DP to Support Vector Machine, which reduced the computation time considerably with only minor loss in accuracy. With regard to deep learning, Krizhevsky [14] learned the fact that the parameters in convolution layer of Convolutional Neural Network (CNN) take only about 5% while the time cost in the layer takes about 95% of whole time. He leveraged DP in convolution layer and reduced time cost effectively. Tencent's Mariana [36] used DP, which gained a 2.67× speed increment with four GPUs. In recent years, more paralleled deep learning methods have been brought up [11]. In the aspect of algorithms, several algorithms have been brought up to accelerate multi-GPU implementation or make the inference more accurate [1, 26] and faster [7, 12]. Moreover, there are researches have been done to integrate DP and MP [8].

3 Proposed method

In this section, we describe the ATE and ACC tasks based on Q&A text pairs and our parallel strategy. On this basis, considering characteristics of Q&A-style reviews, we propose a multi-task model to jointly address the two tasks. Intuitively, the question text tends to be more important, because the aspect term needing categorization tends to appear in the question first. And then, the answer text also involves information related to the aspect term mentioned in the question text. Thus, we need to better model the representation of question text by doing a better job of harnessing relevant aspect information contained in both the question and answer text. Specifically, our proposed model uses two Bidirectional Long Short-Term Memories (Bi-LSTMs) to generate hidden state representations of the question and answer text, respectively. For the ATE task, we use a fully connected layer and CRF layer to extract the aspect term in the question text. For the ACC task, an attention mechanism is applied to capture the most relevant aspect information between the Q&A text, and extend the representation of question text by leveraging the relevant aspect information contained in the answer text. Finally, for making full use of training resources, we design a data parallel strategy for our proposed model.

3.1 Aspect term extraction and aspect category classification tasks

We tackle the ATE task as sequence tagging problem, which extracts an explicit aspect term in the question text. Note that the extracted term could be a single word or a phrase. From the sequence tagging perspective, the word tokens related to the given aspect category should be tagged according to a predefined label scheme. We define the label scheme as {**B**,**I**,**E**,**O**,**S**}, where **B** indicates an aspect term's beginning, **I** indicates the inside of an aspect term, **E** indicates an aspect term's end, **O** means others. In particular, if the aspect term is a single word, we label it as **S**. In this way, the question text "How about the color of the phone?" can be tagged as "How/**O** about/**O** the/**B** color/**I** of/**I** the/**I** phone/**E** ?/**O**". Thus, we address the ATE task by training a sequence labeling model based on the combination of Bi-LSTM and CRF layers.

Instead of a sequence labeling model, the ACC task is considered as a general classification problem. Given the predefined categories, the task is to identify the aspect category for the specified aspect term. Thus, the proposed model uses two Bi-LSTM layers to model representations of the question text and answer text, and then an attention mechanism is adopted to extend the representation of the question text for improving our model's performance on ACC task.

3.2 Multi-task model

Figure 2 shows the architecture of multi-task learning framework. Given a Q&A-style review, assume that the question text $Q = \{w_1, w_2, \ldots, w_M\}$ contains M single words, where w_i represents the ith single word in the question text. Each single word is represented as $q_i \in R^{d_w}$ which is obtained from a word embedding matrix $E \in R^{d_w \times |V|}$, where d_w is the embedding dimension and $|V|$ is the vocabulary size. Thus, we represent the question text as a character-level embedding matrix $S_Q = \{q_1, q_2, \ldots, q_M\}$. Similarly, we represent the answer text $A = \{s_1, s_2, \ldots, s_N\}$ as a character-level embedding matrix $S_A = \{a_1, a_2, \ldots, a_N\}$, where $a_j \in R^{d_w}$ denotes the jth single word of the answer text and N is the number of single words in an answer text.

Next, we feed the character-level embedding matrix of question text S_Q into a Bi-LSTM layer shared by the ATE and ACC tasks to generate a hidden state matrix of question text $H_Q = \{h_{q_1}, h_{q_2}, \ldots, h_{q_M}\}$, where we obtain the hidden state of each single word by averaging the forward and backward hidden state:

$$\overrightarrow{H_Q} = \overrightarrow{LSTM}(S_Q) \tag{1}$$

$$\overleftarrow{H_Q} = \overleftarrow{LSTM}(S_Q) \tag{2}$$

$$H_Q = AVG(\overrightarrow{H_Q}; \overleftarrow{H_Q}) \tag{3}$$

where $H_Q \in R^{d_h \times M}$, d_h is the dimension of hidden state and M is the number of single words in the question text.

3.2.1 Aspect term extraction

Given the hidden state matrix of question text H_Q, the model transforms it into an output label space using an additional fully connected layer:

$$P = H_Q^T \cdot W_{ate} + b_{ate} \tag{4}$$

where $P \in R^{M \times N_t}$ is the output score matrix of N_t labels, and P_{ij} denotes the score of the jth tag of the ith single word in the question text. $W_{ate} \in R^{d_h \times N_t}$ and $b_{ate} \in R^{N_t}$ are parameters of the fully connected layer. Then we leverage conditional random field (CRF) layer for tagging because it takes an object's neighbor into account, which is similar to the use of past and future input features via the bidirectional LSTM layer. The CRF layer takes as an input the sequence of vectors P and returns sequence of labels $z = (z_1, z_2, \ldots, z_n)$. According to the given question text $Q = \{w_1, w_2, w_3, \ldots, w_M\}$, we define the prediction score

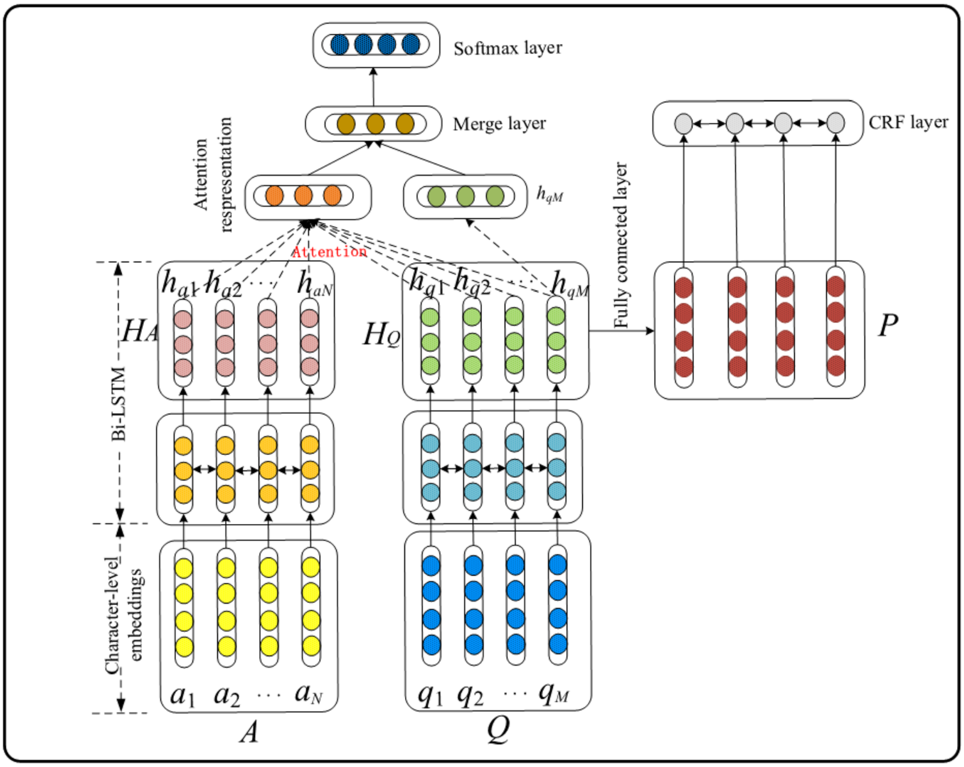


Fig. 2 The architecture of our proposed multi-task model with attention mechanism

for each output tag sequence $z = \{z_1, z_2, \ldots, z_M\}$, where z_i denotes the label of w_i :

$$Score(Q, z) = \sum_{i=1}^{M-1} A_{z_i, z_{i+1}} + \sum_{i=1}^{M} P_{i, z_i} \tag{5}$$

$A \in R^{N_t \times N_t}$ is a transition score matrix, and A_{ij} is the transition probability from label i to label j. Furthermore, we adopt a softmax function over all possible tag sequences for computing the posterior probability:

$$P(z|Q) = \frac{e^{Score(Q,z)}}{\sum_{\tilde{z} \in Z_Q} e^{Score(Q,z)}} \tag{6}$$

where Z_Q denotes all possible tag sequence collections. For CRF training, we use the maximum conditional likelihood estimation. For a training set $\{(\mathbf{Q}_i, \mathbf{z}_i)\}$,the logarithm of the likelihood (a.k.a. the log-likelihood) is given by:

$$L(params) = \sum_i \log p(\mathbf{z}|\mathbf{Q}; params) \tag{7}$$

Maximum likelihood training chooses parameters such that the log-likelihood $L(params)$ is maximized. While decoding, according to the principle of maximizing posterior probability, we select the tag sequence that maximizes the posterior probability as the optimal path z^* and then extract aspect terms according to the tag sequence:

$$z^* = \arg\max_{\tilde{z} \in Z_Q} Score(Q, \tilde{z}) \tag{8}$$

3.2.2 Aspect category classification

Given the hidden state matrix of answer text H_A, the model uses another Bi-LSTM layer to obtain a hidden state matrix of answer text $A = \{s_1, s_2, \ldots, s_N\}$:

$$\overrightarrow{H_A} = \overrightarrow{LSTM}(S_A) \tag{9}$$

$$\overleftarrow{H_A} = \overleftarrow{LSTM}(S_A) \tag{10}$$

$$H_A = AVG(\overrightarrow{H_A}; \overleftarrow{H_A}) \tag{11}$$

where $H_A \in R^{d_h \times N}$ and N is the number of single words in the answer text. Noting that there may be irrelevant aspect terms mentioned in the Q&A text, we adopt an attention mechanism to capture the most relevant aspect information mentioned in both the question and answer context. Thus, the vector representation of question text could be enhanced by making full use of aspect information contained in the answer text. The attention layer calculates the attention representation of the question text according to the following formulas:

$$M = \tanh(W_c \cdot (H_A^T \cdot H_Q) + b_c) \tag{12}$$

$$\alpha = softmax(W_e^T \cdot M) \tag{13}$$

$$r = H_Q \cdot \alpha^T \tag{14}$$

where $M \in R^{N \times M}$, r is the attention representation of question text, and $W_c \in R^{N \times N}$, $b_c \in R^M$, $W_e \in R^N$ are parameters to be trained. Finally, the final vector representation of question text is calculated by non-linearly combining r with the final hidden state h_{q_M} :

$$h^* = \tanh(W_f r + W_x h_{q_M}) \tag{15}$$

where $h^* \in R^{d_h}$, $W_f \in R^{d_h \times d_h}$ and $W_x \in R^{d_h \times d_h}$ are parameters to be trained. In the softmax layer, the final aspect category distribution of the given question text is predicted using the final vector representation of question text h^*:

$$y = softmax(Wh^* + b) \tag{16}$$

where $W \in R^{K \times d_h}$, $b \in R^K$ are parameters in the softmax layer and K is the number of predefined categories.

3.2.3 Data parallel for model

We design our parallelism strategy generally in the following way:

- Divide the model's inputs into multiple sub-batches.
- Apply a model copy on each sub-batch. Every model copy is executed on a dedicated GPU.
- Concatenate the results (on CPU) into one big batch.

As mentioned above, gradients are needed to be passed backward to update parameters within the network. This is normally done by stochastic gradient descent in modern deep learning because the dataset is too big to be fit into the memory. For example, if we have 10K data points in the training dataset, every time we could only use 16 data points to calculate the estimate of the gradients, otherwise our GPU may stop working due to insufficient GPU memories.

The shortcoming of stochastic gradient descent is that the estimate of the gradients might not accurately represent the true gradients of using the full dataset. Therefore, it may take much longer to converge.

A natural way to have more accurate estimate of the gradients is to use larger batch sizes, or even use full dataset. To allow this, the gradients of small batches were calculated on each GPU, the final estimate of the gradients is the the weighted average of the gradients calculated from all the small batches.

Mathematically, data parallelism is valid because of

$$\begin{aligned}\frac{\partial \text{Loss}}{\partial w} &= \frac{\partial\left[\frac{1}{n}\sum_{i=1}^{n} f(x_i, y_i)\right]}{\partial w} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial f(x_i, y_i)}{\partial w} \\ &= \sum_{j=1}^{k}\frac{m_j}{n}\frac{\partial\left[\frac{1}{m_j}\sum_{i=m_{j-1}+1}^{m_j+m_{j+1}} f(x_i, y_i)\right]}{\partial w} \\ &= \frac{m_1}{n}\frac{\partial l_1}{\partial w} + \frac{m_2}{n}\frac{\partial l_2}{\partial w} + \cdots + \frac{m_k}{n}\frac{\partial l_k}{\partial w}\end{aligned} \tag{17}$$

where $m_0 = 0$, w is the parameters of the model, $\frac{\partial \text{Loss}}{\partial w}$ is the true gradient of the big batch of size n, $\frac{\partial l_k}{\partial w}$ is the gradient of the small batch in GPU k, x_i and y_i are the features and labels of data point i, $f(x_i, y_i)$ is the loss for data point i calculated from the forward propagation, n is the total number of data points in the dataset, k is the total number of GPUs, m_k is the number of data points assigned to GPU k, $m_1 + m_2 + \cdots + m_k = n$. When $m_1 = m_2 = \cdots = m_k = \frac{n}{k}$, we could further have:

$$\frac{\partial \text{Loss}}{\partial w} = \frac{1}{k}\left[\frac{\partial l_1}{\partial w} + \frac{\partial l_2}{\partial w} + \cdots + \frac{\partial l_k}{\partial w}\right] \tag{18}$$

Here for each GPU node, we use the same parameters of the model to do the forward propagation, we send a small batch of different data to each node, compute the gradient normally, and send the gradients back to the main node. This step is asynchronous because the speed of each GPU node is slightly different. Once we got all the gradients (we are doing synchronization here), we calculate the (weighted) average of the gradients, and use the (weighted) average of the gradients to update the model/parameters. Then we move on to the next iteration.

3.3 Model training

In the ACC task, given a set of training data S_{Q_t}, S_{At}, and y_t, where S_{Q_t} is the tth question text, S_{At} is the tth answer text, and y_t is the ground-truth aspect category for the Q&A text pair (S_{Q_t}, S_{At}). We use the cross entropy function between y and y_t with L2 regulations as a loss function:

$$L_{acc} = -\sum_{t=1}^{N}\sum_{k=1}^{K} y_t^k \log y^k + \frac{l}{2} \parallel \theta \parallel^2 \tag{19}$$

where N is the size of training set, K is the number of predefined categories, l is the parameter for L2 regularization, and θ is a parameter set.

In the ATE task, given a set of training data S_{Q_t}, S_{At}, and z_t, where S_{Q_t} is the tth question text, S_{At} is the tth answer text, and z_t is the prediction-output tag sequence for the tth question text, assuming Score(S_{Q_t},z_t) is the score of tag sequence z_t, we describe the log-likelihood function as:

$$L_{ate} = \sum_{t=1}^{N} Score(S_{Q_t}, z_t) - \log(\sum_{\tilde{z}\in Z_Q} e^{Score(Q,z)}) \qquad (20)$$

where N is the size of the training set and Z_Q is all possible tag sequence collections. To learn the parameters of the multi-task model, we define the loss function as a weighted linear combination:

$$L = \lambda L_{acc} + (1-\lambda)L_{ate} \qquad (21)$$

where λ is the weight parameter. Parameters are optimized by using Adam optimization functions, and to solve overfitting problems, dropout strategy is adopted.

4 Experiments

4.1 Experimental setting

- **Data settings** Our experiments use Q&A-style reviews as training data which involves in digital domain, beauty domain and luggage domain. Table 1 shows the distribution of experimental data. Each line in the dataset denotes a Q-A pair, and there are three parts in a line: the first part is the question text which is tagged for each character; the second part is the untagged answer text; the third part is a (aspect term, aspect category, sentiment) triple. Considering the problem of imbalanced distribution of data, we discard the aspect categories that involve less than 50 question–answering text pairs.
- **Character-level representations** Considering the informal nature of online reviews, we choose character-level embeddings instead of word-level embeddings to reduce the word segmentation errors. Specially, the character-level embeddings are obtained by using 320 thousand question–answering text pairs extracted from Taobao and we use skip-gram [24] model provided by gensim toolkit to model word respresentations.
- **Evaluation metrics** For aspect category classification task, the main evaluation metrics are Accuracy(A_{acc}) and F1-measure(F_{acc}) where F_{acc} is calculated as $F_{acc} = \frac{2P_{acc}R_{acc}}{P_{acc}+R_{acc}}$. For aspect term extraction task, F_{ate} is calculated by the formula $F_{ate} = \frac{2P_{ate}R_{ate}}{P_{ate}+R_{ate}}$.
- **Hyper-parameters** All out-of-vocabulary words are initialized by sampling from the uniform distribution U(−0.01, 0.01), the dimension of character-level embeddings and hidden state vectors are set to be 300. Other hyper-parameters are tuned according to the development data, the model use Adam optimizer with a batch size 32, and initial learning rate is 0.002. The weight parameters of the multi-task model λ is set to be 0.55, dropout rate is set to be 0.25 to reduce overfitting.
- **Training data percentage** Consider the limitation of training data amount, there is probability that the result of model can still improve if new training data is coming. To test if the model is still under convergence, we do each experiment with different training data percentage from 0.2 to 1, with other settings remain the same.
- **Experiment setup** All experiments were conducted on a GPU cluster. The cluster is equipped with two 12-core Intel(R) Xeon(R) E5-2650 v4 @ 2.20GHz processors and 8 NVIDIA GeForce 1080Ti GPU with 10 GB video Memory of each one.It runs Ubuntu 16.04, CUDA 9.0.176 and CUDNN 7.

4.2 Baseline models

In order to comprehensively evaluate the performance of our multi-task model, we compare our proposed model with several popular baselines for aspect terms extraction and aspect category classification based on quesiton-answering reviews respectively.

In ACC task based on question–answering reviews, we build the following baseline models:

- **LSTM(A)**: This model takes the answering text as input, and uses LSTM network to model the answering text, then the hidden state representations will be used for aspect category classification task.
- **LSTM(Q)** This model takes the question text as input, and uses LSTM network to model the question text,

Table 1 Training data distribution

Statistics	Domain		
	Digital	Beauty	Luggage
Aspect categories	7	11	12
The number of Q&A text pairs	2427	2927	2876
Most frequent aspect category	IO	Efficacy	quality
Q&A text pairs contained most frequent aspect category	908	911	868
Maximum words of aspect term	8	8	7
Minimum words of aspect term	1	1	1

then the hidden state representations will be used for aspect category classification task.

- **LSTM(Q+A)** This model takes question text and answering text as input, and uses LSTM network to model the question context and answering context, then the final hidden state representations is obtained by concatenating the hidden state representations of question text and answering text.
- **Bi-LSTM** This model takes the question text as input, and uses Bi-LSTM network to model the question text, then the hidden state representations will be used for aspect category classification task.
- **Multi-task** This model is a variation of our proposed model for aspect category classification. Compared with ours, it ignores the relevant information between question text and answer text.
- **Multi-task+Attention (MTA)** This proposed model of ours is used for question–answering aspect category classification by constructing a multi-task learning framework. Aspect category classification task is based on Bi-LSTMs with attention mechanism to better represent question text.

In ATE task based on question–answering reviews, we build the following baseline models:

- **CRF** This method uses conditional random fields to extract aspect term from question text. It uses character-level embeddings learned from Skip-gram model as input.
- **Bi-LSTM**: This method uses Bi-LSTM to model question text, and then leverages a softmax layer for aspect term extraction.
- **Bi-LSTM+CRF** This method uses Bi-LSTM to model question text, and feed the hidden states into a CRF layer for aspect term extraction.
- **Mulit-task** This model is a variation of our proposed model for aspect term extraction. Compared with ours, it ignores the relevant information between question text and answer text.
- **Mulit-task+Attention (MTA)** This proposed model of ours is used for question–answering aspect term extraction by constructing a multi-task learning framework. Aspect term extraction task is conducted based on Bi-LSTM and CRF.

4.3 Experimental result and model comparison

Tables 2, 3, 4, 5, 6 and 7 show the performance of proposed model with other baseline models, divided by task of ACC and ATE. Since the curve trends on different domains are similar, we only show and describe the figures of one domain. Experiments are conducted with full training data and single GPU to get a overview of all well-trained models. By analysis, we can draw the following conclusion:

For aspect category classification task, the performace of LSTM(Q) is obviously better than LSTM(A) which proves the idea that question text tends to be more

Table 2 Results of aspect category classification in luggage domain

Model	Precision	Recall	F1	Accuracy
LSTM(Q)	0.6062	0.5571	0.5630	0.6298
LSTM(A)	0.3696	0.3328	0.3220	0.4268
LSTM(Q+A)	0.6525	0.5909	0.5994	0.6656
Bi-LSTM	0.5864	0.5200	0.5354	0.6268
Multi-task	0.6046	0.5660	0.5691	0.6238
MTA(ours)	0.6143	0.5746	0.5774	0.6358

Table 3 Results of aspect term extraction in luggage domain

Model	Precision	Recall	F1
Bi-LSTM	0.6061	0.6477	0.6262
CRF	0.6056	0.2567	0.3605
Bi-LSTM+CRF	0.6125	0.6417	0.6268
Multi-task	0.5863	0.6388	0.6114
MTA(ours)	0.5883	0.6955	0.6374

Table 4 Results of aspect category classification in beauty domain

Model	Precision	Recall	F1	Accuracy
LSTM(Q)	0.5790	0.5356	0.5433	0.6859
LSTM(A)	0.4031	0.3509	0.3579	0.4829
LSTM(Q+A)	0.5631	0.5449	0.5462	0.6844
Bi-LSTM	0.5425	0.5054	0.5085	0.6592
Multi-task	0.5169	0.5011	0.4971	0.6503
MTA(ours)	0.6281	0.5526	0.5646	0.6992

Table 5 Results of aspect term extraction in beauty domain

Model	Precision	Recall	F1
Bi-LSTM	0.5791	0.6503	0.6127
CRF	0.5920	0.3525	0.4419
Bi-LSTM+CRF	0.5891	0.6118	0.6002
Multi-task	0.5941	0.6355	0.6141
MTA(ours)	0.5993	0.5629	0.5806

Table 6 Results of aspect category classification in digital domain

Model	Precision	Recall	F1	Accuracy
LSTM(Q)	0.7996	0.6919	0.7115	0.7883
LSTM(A)	0.4917	0.4492	0.4476	0.5301
LSTM(Q+A)	0.7920	0.7304	0.7421	0.7986
Bi-LSTM	0.7778	0.6706	0.6775	0.7642
Multi-task	0.8140	0.6883	0.7113	0.7745
MTA(ours)	0.8186	0.6906	0.7492	0.7965

Table 7 Results of aspect term extraction in digital domain

Model	Precision	Recall	F1
Bi-LSTM	0.6519	0.5834	0.6158
CRF	0.7173	0.1136	0.1961
Bi-LSTM+CRF	0.6672	0.6454	0.6561
Multi-task	0.6643	0.6471	0.6556
MTA(ours)	0.6293	0.6867	0.6567

important than answering text. Moreover, LSTM(Q+A) outperforms LSTM(Q) and LSTM(A) which inspires us that the combination of question text and answering text could improve the performance of aspect category classification. The multi-task model without attention mechanism achieves the improvement of 3.3%(A_{acc}) and 2.0%(F_{acc}) in digital domain, 1.6%(A_{acc}) and 2.0%(F_{acc}) in beauty domain and 0.9%(A_{acc}) and 1.1%(F_{acc}) in luggage domain which proves that multi-task model could improve the performance of aspect category classification with the help of extracted aspect information. Further, the Multi-task+Attention model achieves the improvement of 2.1%(A_{acc}) and 5.5%(F_{acc}) in digital domain, 3.5%(A_{acc}) and 3.7%(F_{acc}) in beauty domain, 2.8%(A_{acc}) and 1.8%(F_{acc}) in luggage domain which proves that the attention mechanism could capture the most relevant aspect information between question context and answering context and enhance the representation of question text.

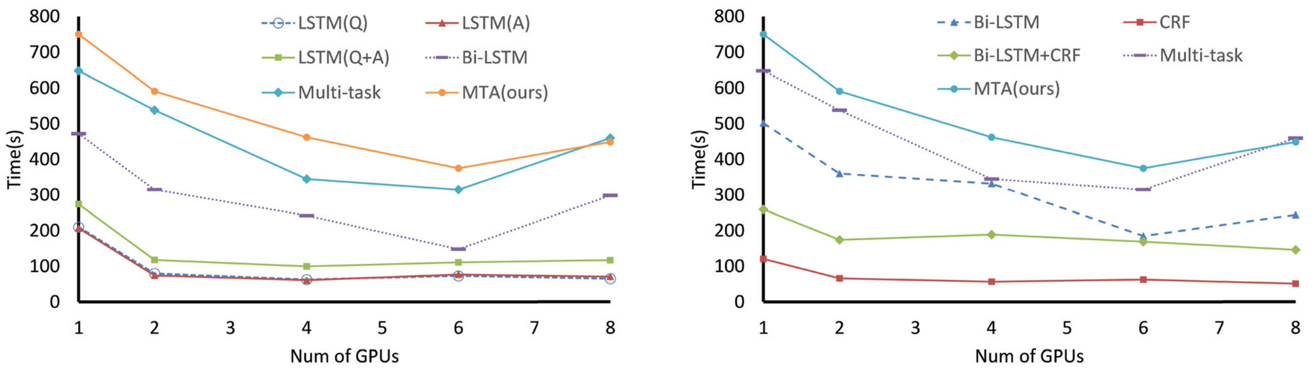
For aspect term extraction task, the model Bi-LSTM+CRF achieves the improvement of 3.2% in digital domain, 0.8% in beauty domain, and 1.5% in luggage domain compared with Bi-LSTM which proves that Bi-LSTM could learn the context information of question text, but softmax layer ignores the interaction of tag sequence. CRF layer introduces state transition matrix which can make use of sentence level tag information to improve the performance of aspect term extraction. Multi-task model without attention mechanism outperforms Bi-LSTM+CRF for the improvement of 1.2% in digital domain, 0.8% in beauty domain and 1.4% in luggage domain which confirms our intuition that aspect category information is helpful to distinguish aspect term from other words unrelated to aspect information. Further, Multi-task+Attention model achieves the improvement of 0.6%, 0.8%, 2.4% in three domains which indicates that the performance improvement of aspect category classification can further enhance the performance of aspect terms extraction.

Simultaneously, to evaluate the affect of DP, we further conduct more experiments.

We first evaluate the efficiency performance of DP by vary the GPU number we use across. Results are shown in Fig. 3. Time includes training time, predicting time and evaluating time. Several conclusions can be drawn from this figure. First, for the fixed mini-batch size 32, total time cost is decreasing in general. For our model, the best result shortens the time by half in digital domain of ACC task. As for others, time shortens 43% and 45% in beauty and luggage domain accordingly. The observation shows the excellent effect of DP. Next, we can notice that the greatest time reduction happens between no parallelism and 2-GPU DP in most cases. This is reasonable because DP's implementation can greatly benefit the model's performance. And with the raise of GPU number, the extent of time reduction reduces, reaching the optimum in 6-GPU cases then begins to rise. As we know, aggregating gradient updates is a critical step in DP. The rise can be explained as the communication overhead has surpassed the time cost reduction brought by distribution.

Comparison of all baselines of ACC task and ATE task accordingly are shown in Fig. 4. We use the optimal GPU number of 6. We can notice that our model reaches the highest F1-measure score in all three domains in both ATE and ACC tasks when trained with 20% of training data, with notable improvements of 7.4% in digital domain, 3.6% in beauty domain and 35.8% in luggage domain respectively for ACC task; 6.2% in digital domain, 7.6% in beauty domain and 6.0% in luggage domain respectively for ATE task. Moreover, the chart lines of our proposed model are flatter than the baselines'. Preceding observations can prove the robustness of our model when the training data is inefficient. Somehow, we may also notice that the result seems still tend to go up when we use all of our training data, which could be a signal that the model is still under its best performance.

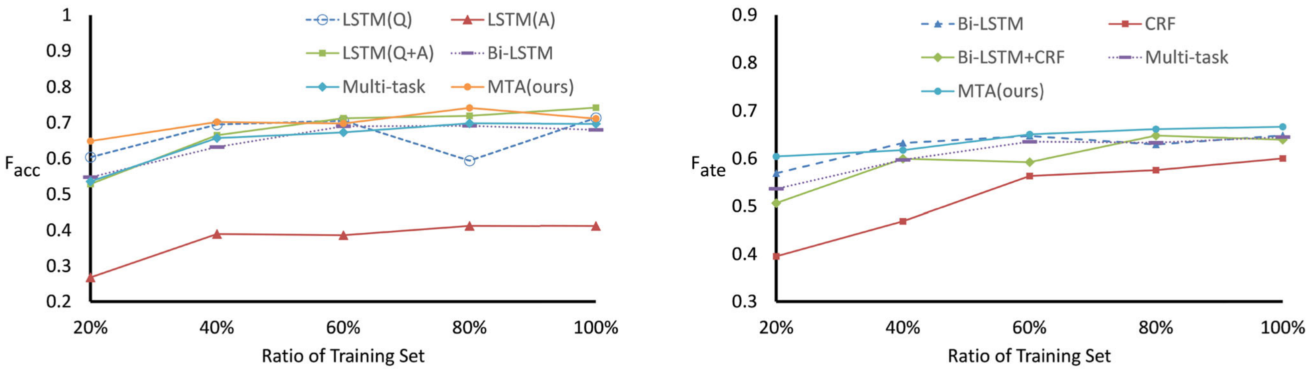
We further conduct experiments on our model with different training set proportion, across different number of GPUs. Certain observations can be found on Fig. 5. First, it's easy to notice that the higher proportion of training data, the better result we got. The result of 20% training data is turbulent through different GPU number, indicates the model's lack of training data. Besides, F1-measure score of other training data proportion behaves rather



(a) ACC task of digital domain

(b) ATE task of digital domain

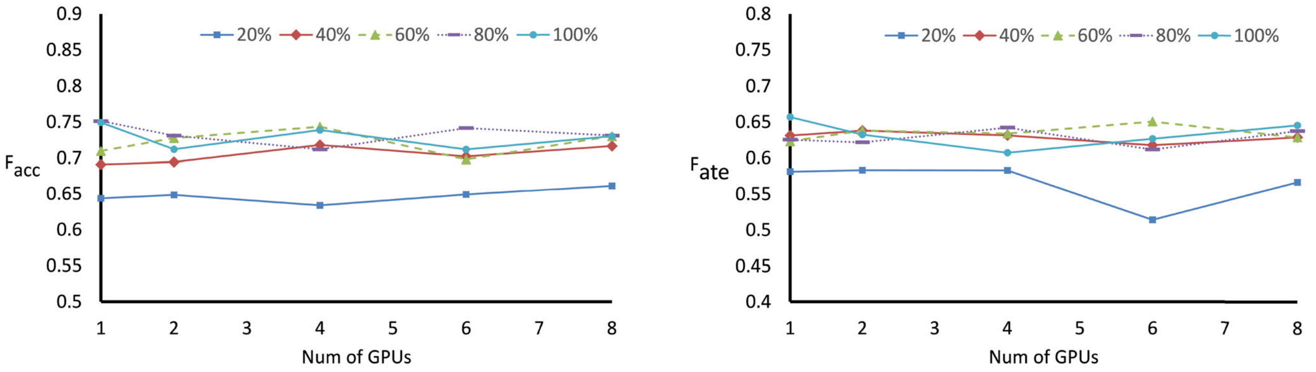
Fig. 3 Time cost of multiple baselines and our model with different GPU utilization number



(a) ACC task of digital domain

(b) ATE task of digital domain

Fig. 4 F1-measure score of baselines trained with different proportions of training data in three domains



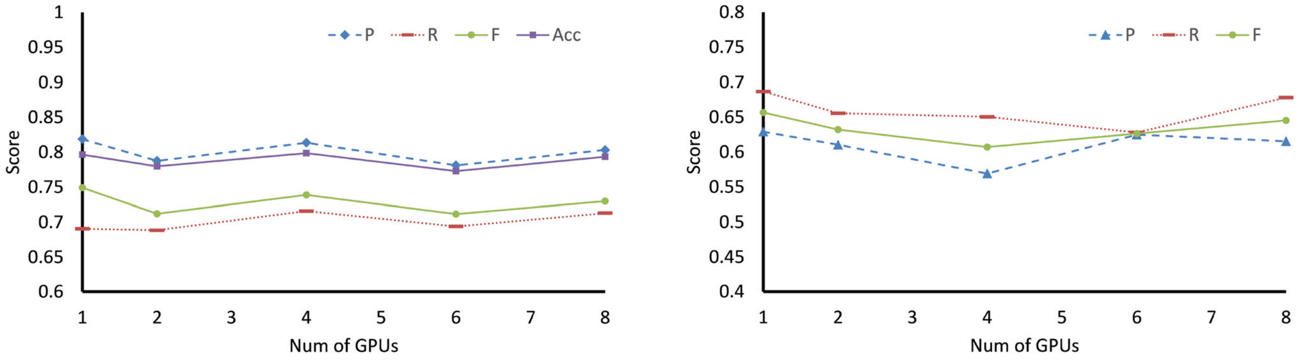
(a) ACC task of digital domain

(b) ATE task of digital domain

Fig. 5 F1-measure score of our model with different GPU utilization number and different training data proportion

stable through all numbers of GPU. At last, we compared precision, recall, F1-measure score and Accuracy of our model differs on number of GPUs we use. Figure 6 shows the trend. we can notice that the result varies little on different evaluation matrix from GPU 1 to 8. The performance has no loss in thus change. Results above can be proofs of our model's robustness.

Experimental results prove that our proposed multi-task model could make full use of the correlation between aspect category classification and aspect term extraction to improve the performance interactively. Besides, the experiment results confirm our two intuitive hypotheses according to characteristics of question–answering style reviews which are that the question texts are more



(a) ACC task of digital domain

(b) ATE task of digital domain

Fig. 6 Precision, recall, F1-measure score and accuracy of our model with increase in the number of GPUs

important than answering texts for aspect category classification task and attention mechanism could capture the most relevant aspect information that is mentioned by both the question text and the answer text to improve the performance of aspect category classification. To end with, the GPU cluster can accelerate model training massively with few negative impacts in our experiments.

4.4 Error analysis

In order to figure out the the limitations of the proposed model, we carefully analyze the misclassified samples in the test set and find the factors that lead to errors as follows. The first factor is imbalanced data distribution which make the model tend to predict the aspect categories that contain more question–answering text. For example, in the digital domain, 22.95% of misclassified samples are predicted to be “IO”. Similarly, in beauty domain and luggage domain, misclassified samples tend to be predict to be “efficacy” and “quality”. The second factor is that in order to reduce word segmentation errors, we choose character-level embeddings rather than word-level embeddings for aspect term extraction and aspect category classification, but modeling only one single word may cause our model unable to model the real semantic/syntactic information of a clause or the whole sentence which results in performance degradation of the subsequent model. The third factor is that the semantic information of some aspect terms are ambiguous in different contexts which lead to the difficulty of aspect category classification. For the data parallelism, we should be aware of that the optimum of GPU utilization number changes with the size of the data. The relationship between data size and the optimum GPU utilization number is remained unclear.

5 Conclusion

In this paper, we propose a multi-task neural learning framework based on question–answering text for addressing aspect category classification and aspect term extraction simultaneously and explore its performance on GPU clusters. The initial inspiration comes from our analysis of characteristics of question–answering style reviews and the correlation between aspect category classification and aspect term extraction, i.e., extracted aspect information can assist aspect category prediction and aspect category information is advantageous to distinguish aspect term from other words unrelated to aspect information. In addition, the motivation of using GPU clusters comes from making full use of advanced training resources, including abundant training data, faster computing speed and more computing power. Experimental results prove that our proposed multi-task model outperforms other baseline models on GPU clusters.

6 Future work

Q&A-style reviews, as a novel form of online review, have great research value. We have achieved some research results in this paper and some previous work [31, 32], but there are still many aspects to be studied and improved. Our future work would like to focus on the followings:

- In order to better model the clause and the whole question sentence, we would like to introduce more complex and powerful pretrained language model to model local contextual information for improving the performance of aspect category classification, because GPU clusters can process and train more larger and complex neural network models in an acceptable training time.

- In question–answering text, in addition to the association between aspect term and aspect category, aspect category and aspect sentiment polarity are also related, thus, we would like to make use of the correlation between aspect category classification and aspect sentiment classification to bulid a joint learning model.
- Considering there may be more than one relevant aspect terms mentioned by both question context and answer context, we would try to conduct aspect category classification for multiple aspect terms simultaneously.

Acknowledgements We would like to thank Dr. Xiong Runqun, Dr. Shen Dian and Dr. Jin Jiahui from School of computer science and engineering of Southeast University, who give us plenty of constructive guidance on modeling and the design of experiments. We would also like to appreciate Jiangsu Provincial Key Laboratory of Network and Information and Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education for offering the environment of our experiments. 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. Alam, M., Perumalla, K.S., Sanders, P.: Novel parallel algorithms for fast multi-gpu-based generation of massive scale-free networks. Data Sci. Eng. **4**(1), 61–75 (2019). https://doi.org/10.1007/s41019-019-0088-6
2. Brychcín, T., Konkol, M., Steinberger, J.: Uwb: machine learning approach to aspect-based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 817–822 (2014)
3. Buck, I., Hanrahan, P.: Data parallel computation on graphics hardware. unpublished report, Jan (2003)
4. Coates, A., Huval, B., Wang, T., Wu, D.J., Catanzaro, B., Ng, A.Y.: Deep learning with COTS HPC systems. In: Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013, pp. 1337–1345 (2013). http://proceedings.mlr.press/v28/coates13.html
5. Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Le, Q.V., Mao, M.Z., Ranzato, M., Senior, A.W., Tucker, P.A., Yang, K., Ng, A.Y.: Large scale distributed deep networks. In: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States, pp. 1232–1240 (2012). http://papers.nips.cc/paper/4687-large-scale-distributed-deep-networks
6. Du, D., Zhang, C., Song, Y., Zhou, H., Li, X., Fei, M., Li, W.: Real-time h control of networked inverted pendulum visual servo systems. IEEE Trans. Cybern. (2019). https://doi.org/10.1109/TCYB.2019.2921821
7. Geng, X., Zhang, H., Zhao, Z., Ma, H.: Interference-aware parallelization for deep learning workload in gpu cluster. Clust. Comput. (2020). https://doi.org/10.1007/s10586-019-03037-6
8. Gholami, A., Azad, A., Jin, P., Keutzer, K., Buluc, A.: Integrated model, batch, and domain parallelism in training neural networks. In: Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures, pp. 77–86 (2018)
9. Hong, X., Li, H., Miller, P., Zhou, J., Li, L., Crookes, D., Lu, Y., Li, X., Zhou, H.: Component-based feature saliency for clustering. IEEE Trans. Knowl. Data Eng. (2019). https://doi.org/10.1109/TKDE.2019.2936847
10. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004, pp. 168–177 (2004). https://doi.org/10.1145/1014052.1014073
11. Kim, Y., Choi, H., Lee, J., Kim, J.S., Jei, H., Roh, H.: Towards an optimized distributed deep learning framework for a heterogeneous multi-gpu cluster. Clust. Comput. (2020). https://doi.org/10.1007/s10586-020-03144-9
12. Kim, Y., Lee, J., Kim, J.S., Jei, H., Roh, H.: Comprehensive techniques of multi-gpu memory optimization for deep learning acceleration. Clust. Comput. (2019). https://doi.org/10.1007/s10586-019-02974-6
13. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: Nrc-canada-2014: detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014, pp. 437–442 (2014). https://www.aclweb.org/anthology/S14-2076/
14. Krizhevsky, A.: One weird trick for parallelizing convolutional neural networks. CoRR arXiv:1404.5997 (2014)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States, pp. 1106–1114 (2012). http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks
16. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001, pp. 282–289 (2001)
17. Li, F., Han, C., Huang, M., Zhu, X., Xia, Y., Zhang, S., Yu, H.: Structure-aware review mining and summarization. In: COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China, pp. 653–661 (2010). https://www.aclweb.org/anthology/C10-1074/
18. Li, K., Chen, C., Quan, X., Ling, Q., Song, Y.: Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7056–7066. Association for Computational Linguistics, Online (2020). https://www.aclweb.org/anthology/2020.acl-main.631
19. Li, X., Bing, L., Li, P., Lam, W., Yang, Z.: Aspect term extraction with history attention and selective transformation. In:

Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, pp. 4194–4200. International Joint Conferences on Artificial Intelligence Organization (2018). https://doi.org/10.24963/ijcai.2018/583

20. Liu, B., Tang, S., Sun, X., Chen, Q., Cao, J., Luo, J., Zhao, S.: Context-aware social media user sentiment analysis. Tsinghua Sci. Technol. **25**(4), 528–541 (2020)
21. Liu, P., Joty, S.R., Meng, H.M.: Fine-grained opinion mining with recurrent neural networks and word embeddings. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17–21, 2015, pp. 1433–1443 (2015). https://www.aclweb.org/anthology/D15-1168/
22. Ma, D., Li, S., Wu, F., Xie, X., Wang, H.: Exploring sequence-to-sequence learning in aspect term extraction. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp. 3538–3547 (2019). https://doi.org/10.18653/v1/p19-1344
23. Meyer, O., Bischl, B., Weihs, C.: Support vector machines on large data sets: simple parallel approaches. In: Data Analysis, Machine Learning and Knowledge Discovery - Proceedings of the 36th Annual Conference of the Gesellschaft für Klassifikation e. V., Hildesheim, Germany, August 2012, pp. 87–95 (2012). https://doi.org/10.1007/978-3-319-01595-8_10
24. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States, pp. 3111–3119 (2013). http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality
25. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., Clercq, O.D., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N.V., Kotelnikov, E.V., Bel, N., Zafra, S.M.J., Eryigit, G.: Semeval-2016 task 5: aspect based sentiment analysis. In: Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16–17, 2016, pp. 19–30 (2016). https://doi.org/10.18653/v1/s16-1002
26. Qin, L., Gong, Y., Tang, T., Wang, Y., Jin, J.: Training deep nets with progressive batch normalization on multi-gpus. Int. J. Parallel Program. **47**(3), 373–387 (2019). https://doi.org/10.1007/s10766-018-0615-5
27. 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, ACL 2017, Vancouver, Canada, July 30–August 4, Volume 2: Short Papers, pp. 148–154 (2017). https://doi.org/10.18653/v1/P17-2023
28. Toh, Z., Su, J.: NLANGP: supervised machine learning system for aspect category classification and opinion target extraction. In: Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4–5, 2015, pp. 496–501 (2015). https://www.aclweb.org/anthology/S15-2083/
29. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Recursive neural conditional random fields for aspect-based sentiment analysis. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016, pp. 616–626 (2016). https://www.aclweb.org/anthology/D16-1059/
30. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4–9, 2017, San Francisco, California, USA, pp. 3316–3322 (2017). http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14441
31. Wu, H., Liu, M., Wang, J., Xie, J., Li, S.: Question–answering aspect classification with multi-attention representation. In: Information Retrieval - 24th China Conference, CCIR 2018, Guilin, China, September 27–29, 2018, Proceedings, pp. 78–89 (2018). https://doi.org/10.1007/978-3-030-01012-6_7
32. Wu, H., Liu, M., Wang, J., Xie, J., Shen, C.: Question–answering aspect classification with hierarchical attention network. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data - 17th China National Conference, CCL 2018, and 6th International Symposium, NLP-NABD 2018, Changsha, China, October 19–21, 2018, Proceedings, pp. 225–237 (2018). https://doi.org/10.1007/978-3-030-01716-3_19
33. Wu, H., Liu, M., Zhang, S., Wang, Z., Cheng, S.: Big data management and analytics in scientific programming: adeep learning-based method for aspect category classification of question–answering-style reviews. Scientific Programming **2020**, 4690974:1–4690974:10 (2020). https://doi.org/10.1155/2020/4690974
34. 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 Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27–December 1, 2017, Volume 2: Short Papers, pp. 151–156 (2017)
35. Yin, Y., Wei, F., Dong, L., Xu, K., Zhang, M., Zhou, M.: Unsupervised word and dependency path embeddings for aspect term extraction. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016, pp. 2979–2985 (2016). http://www.ijcai.org/Abstract/16/423
36. Zou, Y., Jin, X., Li, Y., Guo, Z., Wang, E., Xiao, B.: Mariana: tencent deep learning platform and its applications. PVLDB **7**(13), 1772–1777 (2014). https://doi.org/10.14778/2733004.2733082. http://www.vldb.org/pvldb/vol7/p1772-tencent.pdf

Hanqian Wu is currently an associate professor of School of Computer Science and Engineering and also the director of the Cloud Computing & Big Data lab of College of Software Engineering (Suzhou), Southeast University, China. He received the B.Sc. (1993), M.Sc. (1998) degrees and the Ph.D (2001) degree in Mechanical Manufacturing and Automation from Nanjing University of Aeronautics and Astronautics, China. He is a member of China Computer Federation, his research interests now include Natural Language Processing, Big Data & Machine Learning, Cloud Computing, and Database Technology.

Siliang Cheng received the B.S. degree in Material Science from Tiangong University, China, in 2017. He is currently working toward the M.S. degree in computer science and technology at Southeast University, China. His research interests include Natural Language Processing and Big Data.

Shangbin Zhang received the B.S. degree from Ocean University of China, China, in 2017. He is currently working toward the M.S. degree in computer science and technology at Southeast University, China. His research interests include Natural Language Processing and Big Data.

Zhike Wang received the B.S. degree in computer science from Yancheng Institute of Technology, China, in 2018. He is currently working toward the M.S. degree in computer science and technology at Southeast University, China. His research interests include Natural Language Processing and Big Data.

Feng Yuan received the B.S. degree in mathematics from Soochow University, China, in 2017. He is currently working toward the M.S. degree in computer science and technology at Southeast University, China. His research interests include Natural Language Processing and Data Mining.