Review Rating with Joint Classification and Regression Model

Jian Xu, Hao Yin, Lu Zhang, Shoushan Li*, Guodong Zhou

Natural Language Processing Lab
School of Computer Science and Technology, Soochow University, China
{jxu1017, hyin, lzhang0107}@stu.suda.edu.cn
{lishoushan, gdzhou}@suda.edu.cn

Abstract. Review rating is a sentiment analysis task which aims to predict a recommendation score for a review. Basically, classification and regression models are two major approaches to review rating, and these two approaches have their own characteristics and strength. For instance, the classification model can flexibly utilize distinguished models in machine learning, while the regression model can capture the connections between different rating scores. In this study, we propose a novel approach to review rating, namely joint LSTM, by exploiting the advantages of both review classification and regression models. Specifically, our approach employs an auxiliary Long-Short Term Memory (LSTM) layer to learn the auxiliary representation from the classification setting, and simultaneously join the auxiliary representation into the main LSTM layer for the review regression setting. In the learning process, the auxiliary classification LSTM model and the main regression LSTM model are jointly learned. Empirical studies demonstrate that our joint learning approach performs significantly better than using either individual classification or regression model on review rating.

Keywords: Sentiment Analysis, Review Rating, LSTM

1 Introduction

Sentiment analysis has attracted increasing attention along with the recent boom of e-commerce and social network systems [1]. In sentiment analysis, review rating is a foundational task which aims to automatically assign a score to a review where the score often has a fixed range, such as 1-5 and 1-10. Review rating plays a key role in many real applications, such as recommendation system [2], online advertising [3] and information retrieval [4]. For instance, in a recommendation system, one popular way to recommend a product is to sort all products according to their rating scores which are obtained from the review rating component.

Recently, the leading approaches to review rating deem it as a standard classification problem and have achieved respectable performances in many review rating

* Corresponding author

tasks. For instance, review rating tasks with "5-star" or "10-star" rating systems are considered as a 5-class or 10-class classification problems and we can apply one-vs-all method to return it to several binary classification problems [5]. However, the main criticism of the classification approaches to review rating is that classification approaches do not consider the similarity between class labels. For example, "1-star" is intuitively closer to "2-star" than to "4-star".

Another kind of optional approaches to review rating deem the review rating task as a regression problem which has a natural advantage over the similarity between class labels because of the consideration of different loss between different class labels in the loss function. However, previous studies find that the regression models do not always perform better than classification models. For instance, Pang and Lee [6] empirically show that regression models perform better than classification models in the review rating tasks involving 4 stars but they perform worse than classification models in the review rating tasks involving 3 stars.

Although both the classification and regression models have achieved some success in the study of review rating, most of these methods are built with shallow learning architectures. In recent years, learning methods with deep architectures have achieved significant success in many natural language processing (NLP) tasks, such as machine translation [7], question answering [8] and text categorization [9]. It is a pressing need to extensively exploit the effectiveness of the deep learning methods on the task of review rating.

In this paper, we employ a popular deep learning method, named Long Short-Term Memory (LSTM) network, to perform review rating in terms of both classification and regression models. The main merit of the LSTM method lies in that it equips with a special gating mechanism that controls access to memory cells and it is powerful and effective at capturing long-term dependencies [10].

Furthermore, in order to exploit advantages of both the classification and regression models, we propose a novel approach, namely joint classification and regression model, to review rating. Specifically, we separate the review rating task into a main task (review regression) and an auxiliary task (review classification). An auxiliary representation learned from the auxiliary task with an auxiliary Long Short-Term Memory (LSTM) layer is integrated into the main task for joint learning. With the help of the auxiliary task, our approach boosts the performance of the main task. The experimental result demonstrates that our approach performs better than either the classification LSTM model or the regression LSTM model.

The remainder of this paper is organized as follows. Section 2 overviews related work on review rating. Section 3 presents some basic LSTM approaches to review rating. Section 4 presents our joint classification and regression approach to review rating. Section 5 evaluates the proposed approach. Finally, Section 6 gives the conclusion and future work.

2 Related Work

In the last decade, sentiment analysis has become a hot research area in natural language processing [1]. In this area, review rating is an important task and has attracted more and more attention since the pioneer work by Pang and Lee [6].

One major research line on review rating is to design effective features. Following Pang and Lee [6]'s work, most studies focus on designing effective textural features of reviews, since the performance of a rating predictor is heavily dependent on the choice of feature representation of data. For instance, Qu et al. [11] introduce the bag-of-opinion representation, which consists of a root word, a set of modifier words from the same sentence, and one or more negation words. Beyond textural features, user information is also investigated in the literature of review rating. For instance, Gao et al. [12] use user leniency and product polarity for review rating; Li et al. [13] utilize the textual topic and user-word factors for sentiment analysis. Moreover, polarity shifting is also useful to review rating. For instance, Li et al. [14] propose a machine learning approach to incorporate polarity shifting information into a document-level sentiment classification system. Features learned from other domains can also be useful. For instance, Li et al. [15] propose a multi-domain sentiment classification approach that aims to improve performance through fusing training data from multiple domains.

Another major research line on review rating is to propose novel learning models. Pang and Lee [6] pioneer this field by regarding review rating prediction as a classification/regression problem. They build the rating predictor with machine learning method under a supervised metric labeling framework. Socher et al. [16] introduce a family of recursive neural networks for sentence-level semantic composition. Convolutional neural networks are widely used for semantic composition [17] by automatically capturing local and global semantics. Sequential models like Gated Recurrent Neural Network are also verified as strong approaches for semantic composition [18]. However, it is worthy to note that although these deep learning approaches have been well applied in review rating, they all focus on classification models rather than regression models.

Our work follows the second research line, which aims to propose stronger learning models for review rating. Unlike all above studies, our work is the first to integrate both classification and regression models for review rating and demonstrates that the proposed joint model is a better choice for review rating than using either a classification or a regression learning model.

3 Basic LSTM Models for Review Rating

In this section, we describe some basic LSTM approaches to review rating. The first subsection introduces basic LSTM network. The second subsection delineates the LSTM approach to review classification. The third subsection delineates the LSTM approach to review regression.

3.1 Basic LSTM Network

Long short-term memory network (LSTM) is proposed by Hochreiter and Schmidhuber [10] and it is designed to specifically address this issue of learning long-term dependencies. The LSTM maintains a separate memory cell inside it that updates and exposes its content only when deemed necessary. A number of minor modifications to the standard LSTM unit have been made. In this study, we apply the implementation used by Graves [19] to map the input sequence to a fixed-sized vector.

The architecture of a LSTM unit consists of an input gate i, an output gate o, a forget gate f, a hidden state h, and a memory cell c. At each time step t, the LSTM unit is updated as follows:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + V_i c_{t-1}\right) \tag{1}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + V_f c_{t-1}\right) \tag{2}$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + V_o c_{t-1}\right) \tag{3}$$

$$\tilde{c}_t = \tanh\left(W_c x_t + U_c h_{t-1}\right) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{5}$$

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{6}$$

Where x_t denotes the input at time step t, σ denotes the logistic sigmoid function, $\odot$ denotes elementwise point multiplication. W, U and V represent the corresponding weight matrices connecting them to the gates. Intuitively, the forget gate controls how much the information is discarded in each memory unit, the input gate controls the amount of updated information in each memory unit, and the output gate controls the exposure of the internal memory state.

3.2 Review Rating with LSTM Classification Model

Figure 1 illustrates the classification model architecture for review rating with a LSTM layer. We utilize T^{input} to represent the input, and the input propagates through the LSTM layer, yielding the high-dimensional vector, i.e.,

$$h = LSTM(T^{input}) \tag{7}$$

Where h is the output from the LSTM layer.

Subsequently, the fully-connected layer is applied. The fully-connected layer accepts the output from the previous layer, weighting them and passing through a normally activation function as follows:

$$h^* = dense(h) = \phi(\theta^T h + b) \tag{8}$$

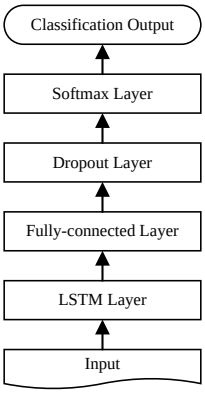

Fig. 1. LSTM classification model

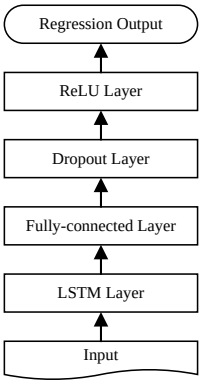

Fig. 2. LSTM regression model

Where ϕ is the non-linear activation function, employed "ReLU" in our model. h^* is the output from the fully-connected layer.

The dropout layer has been very successful on feed-forward networks [20]. By randomly omitting feature detectors from the network during training, it can obtain less interdependent network units and achieve better performance, which is used as a hidden layer in our framework, i.e.,

$$h^d = h^* \cdot D(p^*) \tag{9}$$

Where D denotes the dropout operator, p^* denotes a tuneable hyper parameter (the probability of retaining a hidden unit in the network), and h^d denotes the output from the dropout layer.

The softmax output layer is used for a classification task. The output from the previous layer is then fed into the output layer to get the prediction probabilities, i.e.,

$$p = softmax(W^d h^d + b^d) \tag{10}$$

Where p is the set of predicted probabilities of the review classification, W^d is the weight vector to be learned, and b^d is the bias term.

Our classification model for review rating is trained to minimize a categorical cross-entropy loss function. Specially, the loss function is defined as follows:

$$loss_C = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{l} y_{ij} \log p_{ij} \tag{11}$$

Where $loss_C$ is the loss function of the classification model for review rating, m is the total number of samples, l is the number of review categories, y_{ij} indicates

whether the i-th sample truly belongs to the j-th category, and p_{ij} refers to the predicted probability.

3.3 Review Rating with LSTM Regression Model

Figure 2 illustrates the regression model architecture for review rating with a LSTM layer. From Figure 1 and Figure 2, we can see that most layers, such as the LSTM layer, the fully-connected layer, and the dropout layer, are the same as those in the classification model, which has been described in the last subsection.

Different from the classification model, our regression model utilizes a rectified linear unit output layer instead of a softmax layer, i.e.,

$$f = \mathrm{ReLU}(W^d h^d + b^d) \tag{12}$$

Where W^d and b^d take the same meaning to the classification model above. f is the predicted value, which is a discrete variable.

For the regression model, we employ "mean squared error" for loss function. Specially, the loss function is defined as follows:

$$loss_R = \frac{1}{2m}\sum_{i=1}^{m} \| f_i - y_i \|^2 \tag{13}$$

Where $loss_R$ is the loss function of review regression, y_i is real value and f_i is the predicted value of i-th sample, and m is the total number of the training samples.

4 Review Rating with Joint Classification and Regression Model

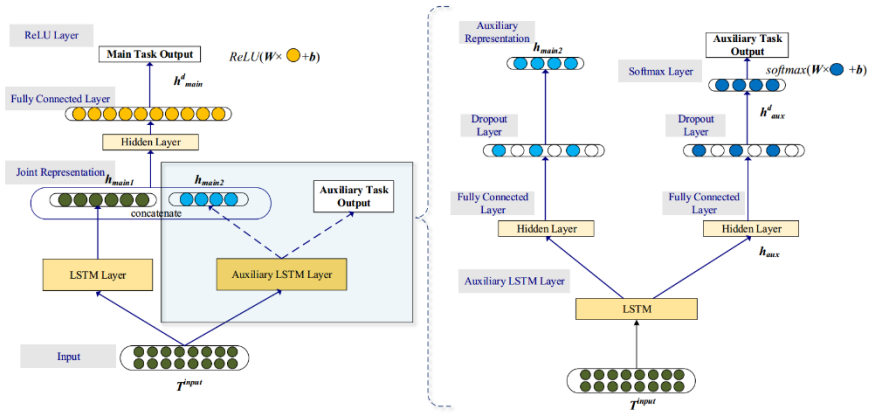

Fig. 3. Overall architecture of the proposed joint-LSTM model for review rating

Figure 3 gives the overall architecture of joint classification and regression model which contains a main LSTM layer and an auxiliary LSTM layer. In our study, we consider the review regression task as the main task and the review classification task as the auxiliary task. The goal of the approach is to employ the auxiliary representation to assist the regression performance of the main task. The main idea of our joint classification and regression approach lies in that the auxiliary LSTM layer is shared by both the main and auxiliary tasks so as to leverage the learning knowledge from both the classification and regression models.

4.1 The Main Task

Formally, the main regression representation of the main task is generated from both the main LSTM layer and the auxiliary LSTM layer respectively:

$$h_{main1} = LSTM_{main}(T^{input}) \tag{14}$$

$$h_{main2} = LSTM_{aux}(T^{input}) \tag{15}$$

The output h_{main1} represents the representation for the regression model via the main LSTM layer and the output h_{main2} represents the representation for the regression model via the auxiliary LSTM layer.
Then we concatenate the two regression representations as the input of the hidden layer in the main task

$$h_{main}^{d} = dense_{main}(h_{main1} \oplus h_{main2}) \tag{16}$$

Where h_{main}^{d} denotes the outputs of fully-connected layer (dense layer) in the main task, and $\oplus$ denotes the concatenate operator.

4.2 The Auxiliary Task

The auxiliary representation is also generated by the auxiliary LSTM layer, which is a reused LSTM layer and is employed to bridge across the classification and regression models. The reused LSTM layer encodes both the same input sequence with the same weights:

$$h_{aux} = LSTM_{aux}(T^{input}) \tag{17}$$

Where h_{aux} represents the representation for the classification model via the reused LSTM layer.

Then a fully-connected layer is utilized to obtain a feature vector for classification, which is the same as the hidden layer in the main task:

$$h_{aux}^{d} = dense_{aux}(h_{aux}) \tag{18}$$

Where h_{aux}^{d} denotes the output of fully-connected layer (dense layer) in the auxiliary task. Other layers including a dropout layer and a softmax layer, as shown in Figure 3, are the same as those which have been described in Section 3.2.

4.3 Joint Learning

Finally, we define our joint cost function for joint classification and regression model as a linear combination of the cost functions of both the main task (i.e., the regression task) and auxiliary task (i.e., the classification task) as follows:

$$loss_{joint-LSTM} = loss_R + loss_C \tag{19}$$

We take RMSprop as the optimizing algorithm. All the matrix and vector parameters in neural network are initialized with uniform samples in $\left[-\sqrt{6/(r+c)}, \sqrt{6/(r+c)}\right]$, where r and c are the numbers of rows and columns in the matrices. In order to avoid over-fitting, the dropout strategy is used to both the main LSTM layer and auxiliary LSTM layer.

5 Experimentation

In this section, we systematically evaluate the performance of our joint classification and regression model for review rating.

5.1 Experimental Settings

Data Settings: Our data are from Mcauley [21] which are collected from Amazon[1]. The data contain 10 domains, i.e., Books, CDs, Phones, Clothing, Electronics, Health, Kitchen, Movies, Sports and Toys. Each domain's ratings range from 1 star to 5 stars. In each domain, we extract a balanced data set from the collected data, i.e., 1000 samples from each star. We use 80% of the data in each review category as the training data and the remaining 20% data as the test data. We also set aside 10% from the training data as the validation data which are used to tune learning algorithm parameters.

Representation: For word representation, we employ skip-gram algorithm (gensim[2] implementation) by word2vec to pre-trained word embedding on the whole data. The length of each text is set to a fixed size.

Basic Prediction Algorithms: (1) Support vector machine (SVM), a popular shallow-learning algorithm, is implemented with the libSVM[3] toolkit. Moreover, we implement SVM regression algorithm with the linear kernel, namely SVR for review

[1] http://Amazon.com/

[2] http://radimrehurek.com/gensim/

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

rating. (2) LSTM, as the basic prediction algorithm in our approach, is implemented with the tool Keras[4]. It is used in both the classifier and regressor for review rating.
Parameters Setting: (1) The parameters of SVM and SVR are set as defaults. (2) The hyper parameters of LSTM are well tuned on the validation data by the grid search method, and most important hyper parameters are shown in Table 1.

Table 1. Parameter setting in learning LSTM

Parameter description	Value
Dimension of the LSTM layer output	128
Dimension of the full-connected layer output	64
Learning rate	0.01
Dropout probability for regression	0.5
Dropout probability for classification	0.5
Epochs of iteration	30

Evaluation Metric: We employ the coefficient of determination R^2 to measure the performance on review rating. Coefficient of determination R^2 is used in the context of statistical models with the main purpose to predict the future outcomes on the basis of other related information. R^2 is a number between 0 and 1. R^2 nearing 1.0 indicates that a regression line fits the data well. Formally, the coefficient of determination R^2 is defined as follows:

$$R^2 = 1 - \frac{SS_{err}}{SS_{tot}} \tag{20}$$

$$SS_{tot} = \sum_i \left(y_i - \bar{y} \right)^2 \tag{21}$$

$$SS_{err} = \sum_i \left(y_i - f_i \right)^2 \tag{22}$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{23}$$

Where y_i is the real value and f_i is the predicted value of each sample.

Significance Test: We randomly split the whole data into training and test data 10 times and employ two different learning approaches, namely A1 and A2, to perform review rating. Then, we employ t-test to perform the significance test to test whether the learning approach A1 performs better than A2 (or otherwise).

[4] https://github.com/fchollet/keras

5.2 Experimental Results

For a thorough comparison, we implement several approaches to review rating. These approaches are introduced as follows.

- **SVM**: The support vector machine classifier with all the parameters default.
- **C_LSTM**: The LSTM classification model which is described is Section 3.2.
- **SVR**: The support vector machine regressor with all the parameters default.
- **R_LSTM**: The LSTM regression model which is described in Section 3.3.
- **AVG_LSTM**: A straightforward approach to integrate the LSTM classification and regression models. Specifically, this approach consists of two main stages. In the first stage, we train a LSTM classifier and regressor respectively. In the second stage, we simply combine the results from the classifier and regressor by averaging them. For instance, if the result of the LSTM classifier is 1 star and the result of the LSTM regressor is 3 star, the combining result is the average of them, i.e., (1+3)/2=2star.
- **JOINT_LSTM**: This is our approach to integrate the LSTM classification and regression models by learning an auxiliary representation for joint learning, which is described in section 4 in detail.

Table 2. Performances of different approaches to review rating

		Books	CDs	Phones	Clothing	Electronics
Classification	SVM	0.115	0.144	0.082	0.087	0.003
	C_LSTM	0.341	0.350	0.235	0.305	0.224
Regression	SVR	0.352	0.411	0.342	0.390	0.333
	R_LSTM	0.502	0.522	0.437	0.500	0.442
Joint	AVG_LSTM	0.506	0.508	0.425	0.488	0.422
	JOINT_LSTM	**0.540**	**0.557**	**0.455**	**0.520**	**0.466**
		Health	Kitchen	Movies	Sports	Toys
Classification	SVM	0.077	0.278	0.200	0.269	0.307
	C_LSTM	0.306	0.377	0.347	0.372	0.440
Regression	SVR	0.346	0.458	0.399	0.437	0.460
	R_LSTM	0.444	0.554	0.481	0.529	0.559
Joint	AVG_LSTM	0.465	0.540	0.495	0.524	0.557
	JOINT_LSTM	**0.472**	**0.565**	**0.520**	**0.542**	**0.572**

Table 2 shows the performances of different approaches to review rating. From this table, we obtain following findings.

Regression models perform much better than classification models. Specifically, SVR outperforms SVM with a wide margin and R_LSTM consistently outperforms C_LSTM with a wide margin. This is mainly due to the fact that the regression models are more suitable for the evaluation metric R^2. Significance test shows that SVR significantly outperforms SVM (p-value<0.001) and R_LSTM significantly outperforms C_LSTM (p-value<0.001).

In regression models, R_LSTM performs much better than SVR in all 10 domains. This result encourages to apply deep learning approaches to the task of review rating. Significance test shows that R_LSTM significantly outperforms SVR (p-value<0.001).

When combining the LSTM classification and LSTM regression models, AVG_LSTM performs well in some domains, such as Books, Health, and Movies, achieving better R^2 than R_LSTM. However, in some other domains, such as CDs, Phones, and Clothing, it performs worse than R_LSTM. These results demonstrate th at simply averaging the results of C_LSTM and R_LSTM would not always improve the performances of R_LSTM.

When combining the LSTM classification and LSTM regression models, JOINT_LSTM outperforms R_LSTM in all 10 domains. Averagely, JOINT_LSTM improves R_LSTM with about 0.024 in R^2. This result verifies the effectiveness of the proposed joint model to review rating. Significance test shows that our approach, i.e., JOINT_LSTM, significantly outperforms both R_LSTM and AVG_LSTM (p-value<0.01).

6 Conclusion

In this paper, we propose a novel approach, namely joint classification and regression model, to review rating, by exploiting advantages of both the review classification and regression models. In our approach, we employ an auxiliary LSTM layer to learn the auxiliary representation in the review classification task (as the auxiliary task) and employ it in the regression task (as the main task). To achieve this, a neural network based model, namely joint LSTM, is employed to bridge across the classification and regression models via a shared LSTM layer. Empirical studies demonstrate that the LSTM model is appropriate for both the review classification and regression task. Moreover, the results show that our joint learning approach significantly boosts the performance of the main regression task in all 10 domains.

In our future work, we would like to improve joint LSTM by looking for better classification models for review rating. Furthermore, we would like to apply our proposed joint LSTM model in other NLP applications which involve both the classification and regression implementations.

Acknowledgments

This research work has been partially supported by three NSFC grants, No.61375073, No.61672366 and No.61331011.

References

1. Pang B, Lee L. Opinion mining and sentiment analysis[J]. Foundations and Trends® in Information Retrieval, 2008, 2(1–2): 1-135.
2. Yang D, Zhang D, Yu Z, et al. A sentiment-enhanced personalized location recommendation system[C]//Proceedings of the 24th ACM Conference on Hypertext and Social Media. ACM, 2013: 119-128.

3. Fan T K, Chang C H. Sentiment-oriented contextual advertising[C]//European Conference on Information Retrieval. Springer, Berlin, Heidelberg, 2009: 202-215.
4. Zhang M, Ye X. A generation model to unify topic relevance and lexicon-based sentiment for opinion retrieval[C]//Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2008: 411-418.
5. Rifkin R, Klautau A. In defense of one-vs-all classification[J]. Journal of machine learning research, 2004, 5(Jan): 101-141.
6. Pang B, Lee L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales[C]//Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics, 2005: 115-124.
7. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.
8. Iyyer M, Boyd-Graber J L, Claudino L M B, et al. A Neural Network for Factoid Question Answering over Paragraphs[C]//EMNLP. 2014: 633-644.
9. Zhang M L, Zhou Z H. Multilabel neural networks with applications to functional genomics and text categorization[J]. IEEE transactions on Knowledge and Data Engineering, 2006, 18(10): 1338-1351.
10. Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
11. Qu L, Ifrim G, Weikum G. The bag-of-opinions method for review rating prediction from sparse text patterns[C]//Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, 2010: 913-921.
12. Gao W, Yoshinaga N, Kaji N, et al. Modeling User Leniency and Product Popularity for Sentiment Classification[C]//IJCNLP. 2013: 1107-1111.
13. Li F, Wang S, Liu S, et al. SUIT: A Supervised User-Item Based Topic Model for Sentiment Analysis[C]//AAAI. 2014, 14: 1636-1642.
14. Li S, Lee S Y M, Chen Y, et al. Sentiment classification and polarity shifting[C]//Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics, 2010: 635-643.
15. Li S, Zong C. Multi-domain sentiment classification[C]//Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers. Association for Computational Linguistics, 2008: 257-260.
16. Socher R, Perelygin A, Wu J Y, et al. Recursive deep models for semantic compositionality over a sentiment treebank[C]//Proceedings of the conference on empirical methods in natural language processing (EMNLP). 2013, 1631: 1642.
17. Johnson R, Zhang T. Effective use of word order for text categorization with convolutional neural networks[J]. arXiv preprint arXiv:1412.1058, 2014.
18. Tang D, Qin B, Liu T. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification[C]//EMNLP. 2015: 1422-1432.
19. Graves A. Generating sequences with recurrent neural networks[J]. arXiv preprint arXiv:1308.0850, 2013.
20. Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors[J]. arXiv preprint arXiv:1207.0580, 2012.
21. McAuley J, Pandey R, Leskovec J. Inferring networks of substitutable and complementary products[C]//Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015: 785-794.