Journal of Intelligent & Fuzzy Systems 38 (2020) 4915–4927
DOI:10.3233/JIFS-191572
IOS Press

A parallel neural network structure for sentiment classification of MOOCs discussion forums

Yi Gao[a], Xia Sun[a,*], Xin Wang[a], Shouxi Guo[a] and Jun Feng[a,b,*]
[a]*Department of Computer Science, Northwest University, Xi'an, China*
[b]*State-Province Joint Engineering and Research Center of Advanced Networking and Intelligent Information Services, Northwest University, Xi'an, China*

Abstract. Forum posts in Massive Open Online Courses (MOOCs) support an important way for online learners to interact with each other and with instructors. Instructors explore the sentiment from posts in MOOCs to detect learners' trending opinions towards the course so that they can improve MOOCs. However, it is unrealistic to expect instructors to adequately track learners' sentiment under the large number of messages exchanged on the forums. Fortunately, sentiment classification can automatically analyze learners' emotion on the course of MOOCs from posts. Traditional classifiers based on machine learning algorithm, which often depend on human-designed features and have data sparsity problem. In contrast to traditional approaches, we develop a novel neural network model called parallel neural network (PNNs) for sentiment classification of MOOCs discussion forum to alleviate the aforementioned problems. In our model, we design a parallel neural network structure to replace the popular serial neural network structure so that PNNs can preserve the validity of features as far as possible when neural network model training. Meanwhile, we also introduce Self-attention mechanism that automatically identifies which features play key roles in sentiment classification to obtain the important components in posts. We experiment on a public MOOCs dataset and two common sentiment classification datasets, and achieve a good performance. That means PNNs is a substantially reliable classification model for identifying the sentiment polarity of posts. The study has great potential application value on the platform of large scale courses, which can help instructors to gain the emotional tendency of learners for the course content in real time, so that timely intervention to support learning and may reduce the dropout rates.

Keywords: Parallel neural network, sentiment classification, MOOCs, learners' sentiment

1. Introduction

Massive Open Online Courses (MOOCs) are a teaching platform that offers free education resources for large-scale online learners. MOOCs continually attracts numerous learners from the world because of unconstrained time and space [13, 19, 32]. According to statistics, more than 58 million users registered at least one course on MOOCs [4], and in excess of 700 universities are providing thousands of courses that distribute on different platforms such as Coursera and edX [36]. In MOOCs, discussion forums play an important role for online learners to mutual communicating and studying [4, 25]. Furthermore, there may be many leaners discussing some contents of a course on the discussion forum, and also expressing their opinions about a course [25, 37].

Working towards improving MOOCs, it is fundamental to know opinions of learners about the course and the major course tools as well [45]. Nicely, the MOOCs discussion forum posts make it possible for instructor teams to understand the learner's learning states and their feelings of the course. However, the total number of forum reviews (these reviews are mainly text) about a course can be very large so that it is infeasible to read and analyze all of them individually [28]. Sentiment classification can reveal

*Corresponding author. Xia Sun and Jun Feng, Department of Computer Science, Northwest University, Xian 710127, China. E-mails: raindy@nwu.edu.cn (Xia Sun) and fengjun@nwu.edu.cn (Jun Feng)

various affective trends according to recent research on social media [45]. Sentiment classification, as one of a fundamental task in the field of Natural Language Process (NLP), offers an opportunity for instructor teams to gain an insight into how learners feel with the course so that they have the ability to modify the course based on the comments of learners to improve their engagement and satisfaction, which is very important to ensure the success of the MOOCs [30].

Wen et al. classify reviews' sentiment by the sentiment lexicon [45]. Classified accuracy of this method relies on the quality of sentiment lexicon and emotional word extraction conditions, and both require manual design. Pedro et al. use some machine learning algorithms, such as the Logistic Regression (LR), the Support Vector Machine (SVM) etc. to achieve sentiment classification in MOOCs [28]. They usually use the bag-of-word (BoW) to represent features [2, 22]. However, due to the data sparsity problem heavily affects the classification accuracy of machine learning approaches [22, 41]. Recently, the rapid development of neural networks (NNs) has brought new inspiration to solve the data sparsity problem [7, 15, 26, 27, 40]. NNs avoid suffering from this problem via using word embedding to a certain extent, and can automatically capture meaningful features during the training process [34]. Therefore, we adopt a NNs approach to perform sentiment classification task of MOOCs in this paper, and expect to achieve a better classification result than previous research.

Although research on sentiment classification of MOOCs use NNs has not been published at present, this method has been widely used in the sentiment classification of other fields in NLP, such as E-commerce websites, stock forecast, political orientation analyses [46]. Convolutional Neural Networks (CNNs) [24] and Long Short-term Memory (LSTM) [14] are two widely used NNs for sentiment classification. CNNs are better at automatically extracting local features from text vectors [17, 20, 21]. As another popular network, LSTM is a strong sequential model to deal with sequence data, which is designed to capture long-distance dependence information from text vectors [23, 40, 48]. In order to let model both have the advantages of CNNs and LSTM, some hybrid combination methods are proposed to capture features and dependency information from texts. The serial combinations are usually adopted, which means LSTM directly stacked on CNNs or reverse [16, 41, 49].

For aforementioned approaches, although CNNs play an extremely strong role to extract local semantic features of sentences, LSTM could reduce the effectiveness of these features extracted by CNNs when these semantic features are sent to LSTM in order to capture their dependence information. So these serialization composite NNs fail to achieve the desired sentiment classification results. The reason of the above problem is that LSTM is a biased model [22], where later features are more dominant than earlier features. But the key components could be any features in all features rather than at the end. Under this umbrella, some features play a major role in sentiment classification may be ignored to reduce the model classification effect. To address the limitation of serialization composite models, we propose a parallel combination way (parallel neural network structure) to preserve the effectiveness of features extracted by CNNs and LSTM as far as possible and amplify the benefits of these two NNs respectively.

We obtain semantic features of a post with long-distance dependence information automatically via a parallel neural network structure of CNNs and LSTM. But these features are directly used to classify that is unreasonable, because each feature contributes equally for the classification now. It is well-known that a feature, containing more emotional information, is more important for sentiment classification. Therefore, the more useful features should be selected for classifying to improve the classification accuracy of the model. Hence, self-attention mechanism [43] is introduced into our paper to feature selection. It can achieve the automatic selection of features without human-selected features via increasing the useful features' weight. In addition, self-attention mechanism could capture global structural information of sentences [43, 47] and conducts direct connections between two features, thus distant features could be calculated by shorter paths (the calculation distance is $O(1)$).

To verify the effect of our model, we conducted experiments on a MOOCs public dataset, namely the Stanford MOOCs posts dataset (it can be availible at: http:// datastage.stanford.edu/StanfordMoocPosts/). The experiments demonstrate that our method has a good performance on this public dataset. Overall, the main contributions of this work are as follows:

1. We design a novel parallel neural networks, namely PNNs, to automatically extract semantic features with dependent and structural information for sentiment classification to achieve better sentiment classification results in MOOCs. Moreover, we believe that it is the first

time using NNs for identifying the sentiment polarity of posts in MOOCs discussion forums.
2. Differing from pioneering hybrid NNs models adopted serialization combination structure, we propose a parallel structure of CNNs and LSTM that is one of components of PNNs, which can enable PNNs prevent these local semantic features extracted by CNNs from affected by LSTM to remain the integrality of all features compared with the previous popular serialization models. According to the experimental results show that the F_1 score of PNNs with this structure improves by 1.16%.
3. Self-attention mechanism is also a component of PNNs, it can directly connect between two arbitrary features of a sentence, which allows unimpeded information flow through the network and assigns different weight coefficient for different features accordingly their importance. Therefore, we can obtain structural information of a sentence and achieve features selection, which is beneficial for classification to further enhance the effect of PNNs.

2. Related work

The serious dropout rate of learners is a problem for MOOCs since the opening of MOOCs, hence, many researchers would like to find the factors which affect the learning of participants to reduce the attrition. Mackness et al. [1] raise a question of how to design a MOOCs that can provide learners with positive experience to increase the completion rate of them. Some the prior work solve this question by surveys and interviews [3, 35]. However, these ways are too time consuming to keep track of the student's learning sentiment in real time. After that, research turns around automatically classifying the sentiment of learners through comments from learners [8, 11, 38]. Sentiment is important to monitor since learners with a positive emotion have been demonstrated to be more motivated in MOOCs settings [29]. Some researchers implement the sentiment classification of MOOCs posts by the sentiment lexicon [9, 33, 45], whose classification accuracy depends on sentiment lexicon quality. Recently, machine learning algorithms, that need not sentiment lexicon, are often used to classify learners' sentiment in MOOCs [5, 28, 31], where SVM has the best classification effect [28]. Nonetheless, these machine learning methods have the data sparsity problem, and their features for classification are human-designed.

Recently, the approaches based on NNs to deal with sentiment classification gradually become popular in NLP. Word embedding extremely alleviates the data sparsity problem [6], and as the input of NNs can improve NNs model effect. CNNs perform successfully in sentiment classification because of capturing local features automatically and efficiently [7, 21, 41]. Due to its ability of processing sequence data and extracting syntax features is widely used for sentiment classification task [16, 23]. Some researchers are expected to obtain the advantages of these two NNs, using a serial structure to combine the two networks to improve the accuracy of sentiment classification [41]. Huang et al. use a CNN layer to capture features and feed the features to two-layer LSTMs, which can extract context-dependent features for sentiment classification [16]. Zhang et al. combine CNNs and LSTM to obtain local and temporal information for sentiment classification [49]. However, the all aforementioned approaches ignore the fact that LSTM is a discriminatory NNs model. In this paper, we propose a parallel structure to combine CNNs and LSTM so that prevent LSTM from affecting features extracted by CNNs.

3. Methods

The sentiment classification in this study focuses on identifying the sentiment polarity of learners in posts, that is, whether these are either positive or negative posts. PNNs is designed to achieve this goal, and Fig. 1 presents its network structure. The input of the network is a post D, which is a sequence of word vectors $w_1, w_2, \ldots, w_l$. These words are represented by word embedding. Then, we use parallel neural network structure showed in Fig. 1 to acquire semantic features with long-distance dependence information, which will be specifically illustrated in Subsection 1. After that these features are sent to Self-attention mechanism for selecting feature and capturing global structural representation of sentences. Finally, the classification layer predicts the sentiment polarity of D by PNNs. In this section, we will introduce our method in detail.

3.1. Parallel neural network structure for feature extraction

We propose a parallel neural network structure to capture semantic and dependence information of

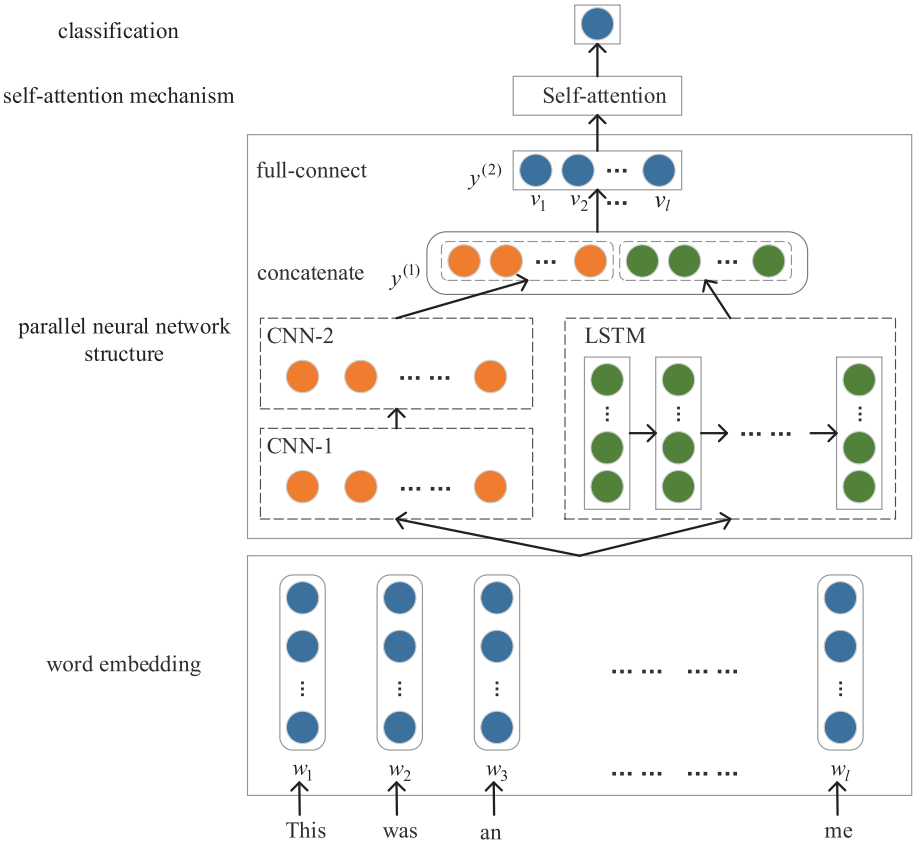

Fig. 1. The structure of PNNs. This figure shows an example of the post "*This was an awesome analysis. Now probability started making sense to me.*" *and the subscripts denote the position of the corresponding word in the original document.*

words from a post, which can maximize the validity of the semantic information extracted by CNNs and long-distance dependence information captured by LSTM. Here, the two-layer CNNs (the first one is called CNN-1, and another one is CNN-2 showed in Fig. 1) are used to extract deep semantic information of sentences, and the LSTM is used to capture long-distance dependence information of sentences. Then the semantic information and long-distance dependence information are combined by concatenate layer, and whose results are sent to full-connect layer to merge two information. By this structure, we can obtain semantic features with dependent information of each post.

We first describe word representation before introducing parallel neural network structure for semantic and denpendent features extraction. We use word2vec toolkit provided by Google to present each word's vector of $D = [w_1, w_2, \ldots, w_l]$, where we define w_i as the i_{th} word in D. l_w is the dimension of w_i and l is vocabulary size of a sentence, so the input post $D \in \mathbb{R}^{l_w \times l}$. After that, each word vector w_i is represented as a low dimensional, continuous and real-valued vector by word embedding layer (showed in Fig. 1). Then, D will be represented as a word embedding matrix D_e showed in Equation (1).

$$D_e = [e(w_1), e(w_2), \ldots, e(w_l)] \tag{1}$$

Where $e(w_i)$ is the embedding representation of w_i, and $|e|$ is the dimension of word embedding, so $D_e \in \mathbb{R}^{|e| \times l}$.

From the word embedding layer, we can obtain the basic word embedding matrix D_e for each post, and then it will be sent to parallel neural network structure which consists of two-layer CNNs, a LSTM, a concatenate layer, and a full-connect layer. In our paper, we use it to extract semantic and long-distance dependence information of whole text.

It is well-known that convolution operations performed by different convolution kernels can extract different text features. Therefore, the CNN-1 is used to extract shallow and local semantic features of the post, and it is calculated by Equation (2):

$$con_j = f[W_1 \cdot e(w_{i:(i+l_c-1)}) + b_1] \quad (2)$$

Where s_1 denotes the number of filter, l_c is the window size of filter. $W_1 \in \mathbb{R}^{s_1 \times |e|}$ is the weight parameter, $b_1 \in \mathbb{R}^{s_1}$ is a bias term, and f is the non-linear activation function (here we use $ReLu$ as the activation function).

After a filter glides through the whole sentence with the form of a window, which produces the total shallow semantic feature vector F_{con}. In order to enable the length of the sentence unchanged after the CNN-1 layer operation and convenient for subsequent operations, we chose the same padding as the padding method for this convolution. So $F_{con} \in \mathbb{R}^{(l \times s_1)}$ is described as Equation (3):

$$F_{con} = [con_1, con_2, \ldots, con_l] \quad (3)$$

Next, the shallow semantic feature vector F_{con} is sent to the CNN-2 layer, stacked on CNN-1 layer directly (showed in Fig. 1), to extract deeper semantic information than CNN-1 layer. The operation of CNN-2 is similar with CNN-1 apart from the different parameters s_2, W_2, b_2, where s_2 presents the filter size of CNN-2. And its calculation method is expressed by Equation (4), then we can obtain the deep semantic feature vector $F_{D\_con} = [D\_con_1, D\_con_2, \ldots, D\_con_l]$.

$$D\_con_j = f[W_2 \cdot con_{i:(i+s_2-1)} + b_2] \quad (4)$$

Capturing long-distance dependence relationship between words in sentences is critical to improving classification accuracy, because words are interdependent. Compared with simple RNN model, LSTM can address the problem of gradient disappearance to a certain extent and can effectively learn sequence characteristics. Meanwhile, LSTM shows better performance on capturing long-distance dependence information from text than CNNs so that LSTM is used to do that. Differing from the pioneers, we adopt CNNs and LSTM in a parallel combination, which means that LSTM (showed in Fig. 1) directly captures long-distance dependence information from D_e. By this way, we not only obtain the significant long-distance dependence information of text, but also prevent the deep semantic features extracted by two-layer CNNs from being negatively affected by LSTM. Here, we define d_i as the denpendent feature of each word extracted by LSTM and its expression as Equation (5). Finally, we obtain the dependent feature matrix F_{LSTM} of a sentence (showed in Equation (6)).

$$d_i = LSTM(e(w_i)) \quad (5)$$

$$F_{LSTM} = [d_1,\ d_2,\ \ldots,\ d_l] \quad (6)$$

Where $d_i \in \mathbb{R}^h$, $F_{LSTM} \in \mathbb{R}^{l \times h}$, h is the number of computational units in the LSTM layer. Both F_{D_con} and F_{LSTM} are combined by Equation (7) to obtain $y^{(1)} \in \mathbb{R}^{l \times |s_2+h|}$ that contains local deep semantic features and long-distance dependence features after we obtain these meaningful features extracted by aforementioned operations.

$$y^{(1)} = [F_{D\_con};\ F_{LSTM}] \quad (7)$$

In order to obtain the final semantic feature $y^{(2)}$ called contextual semantic features, the full-connect layer has been applied for feature fusion. Meanwhile, this operation will not only combine deep semantic feature F_{D_con} and dependent feature F_{LSTM} into an organic entirety, but reduce the dimension of $y^{(1)}$. By this way, we can avoid suffering from the separation of semantic features and long-distance dependence information that is good for improve our results, using Equation (8) to combine the two types feature.

$$y^{(2)} = ReLu(W_3 y^{(1)} + b_3) \quad (8)$$

3.2. Self-attention mechanism for feature selection

After obtaining the contextual semantic features $y^{(2)}$ of the input post, Self-attention mechanism is adopted to learn the weight coefficient of each feature in $y^{(2)}$.

Self-attention mechanism is a resource allocation method, which can ignore the secondary information from the vast amount of features and pay attention on the important information in the sentence. The earlier point is reflected in the weight coefficient, which indicates the importance of each feature. In other words, the more important a feature is in a sentence, the greater its weight coefficient is. Self-attention mechanism enhances the proportion of key features by assigning different weights for different features according to their importance degree. This can effectively help us to reduce the loss of key features during model training process to achieve the purpose of feature selection automatically and obtaining structural information of a sentence.

Here, Fig. 2 shows the inner architecture of Self-attention mechanism in Fig. 1. After obtaining contextual semantic features $y^{(2)} = [v_1, v_2, \ldots, v_l]$, then transported to Self-attention mechanism in Fig. 2 for learning the new representation of sentence by Equation (9):

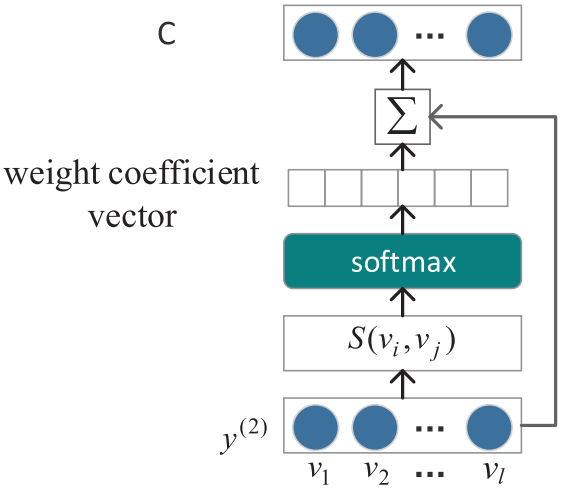

Fig. 2. Self-attention mechanism.

$$C(v_i) = \sum_{j=1}^{l} \beta_{ij} v_i \tag{9}$$

Meanwhile, it is noteworthy that Self-attention mechanism occupies a small proportion of training cost, because it calculates the dependency relationship between any two elements directly regardless of the distance of these (each calculation only needs $O(1)$). In addition, it computes weight coefficient of each element from the whole sentence, so we can gain the structural information by it. More specifically, the weight coefficient β_{ij} of each v_i is computed by the following Equations:

$$\beta_{ij} = \frac{exp(v_{ij})}{\sum_{k=1}^{H} exp(v_{ik})} \tag{10}$$

$$v_{ij} = S(v_i, v_j) \tag{11}$$

Where v_{ij} represents the score about the degree of dependency between the factors v_i and v_j, S is a dot-product to compute the score about two factors. The weight coefficient β_{ij} is obtained via sending v_{ij} to *Softmax* function to compute shown in Fig. 2 by Equation (10). Therefore, the weight coefficient matrix β is acquired. Finally, we get the final representation of a sentence $y^{(3)} = [C(v_1), C(v_2), \ldots, C(v_l)]$.

3.3. Classification layer

The final layer of our model is the output layer, whose output result is the probability of classifying the post as positive or negative. Here, we adopt a linear function with *sigmoid* activation function to get the class probability.

$$p = sigmoid(W_4 y^{(3)} + b_4) \tag{12}$$

Where $y^{(3)}$ refers to the final representation obtained by the above operations. $W_4 \in \mathbb{R}^{1\times H}$ and $b_4 \in \mathbb{R}^1$ are the parameters of PNNs model need to be updated during training, where H is hidden layer size. p ($p \in [0, 1]$) refers to the probability that distinguishes the post belongs to a positive or negative category, when $p \geq 0.5$, the post's category is positive, otherwise it is negative.

4. Experimental design

In this section, we use the Stanford MOOCs posts dataset to our model, and we compare the effect of our proposed model with other previous popular approaches to verify whether PNNs is a generalizable and robust model and can classify the sentiment polarity of per post correctly.

4.1. Datasets

The Stanford MOOCs posts dataset is a large-scale text dataset, and includes 29,604 posts collected from three domains, namely Humanities, Medicine, and Education. Sentiment score of each post was coded by humans so that the data provides a ground truth for the proposed model (more information about score computation can be found in the dataset website. In our analysis, there are 29,570 posts in the dataset for experiment after we excluded posts that contain exception symbols. In the sentiment dimension, coders ranked the sentiment of the post on a scale of 1-7. A score of 7 means the learner who posted the post is very satisfied with the course, while 1 means the learner is extremely unsatisfied and requires immediate attention from the instructors. In this paper, we define the sentiment classification task in MOOCs as a binary classification because our goal is to determine whether the post is positive or negative. We consider these posts are positive whose sentiment *score* ≥ 4, otherwise the posts are negative. Moreover, the full dataset will be split as three subdatasets according to the domain of posts for building model respectively. Table 1 provides detailed information about each subdataset, such as the number of subdataset and its sentiment distribution.

Moreover, in order to verify the availability and effectiveness of the PNNs, we also performed experiments on other common sentiment classification datasets – IMDB-2 and IMDB-10 datasets. We use

Table 1
Properties statistical information of three subdatasets

Domain	Size	Sentiment	
		Positive	Negative
Humanities	9,698	8,853	845
Medicine	9,994	8,145	1,849
Education	9,878	8,189	1,689

Table 2
Statistical information of IMDB-2 and IMDB-10 datasets

Dataset	Train size	Test size	Class
IMDB-2	40,000	10,000	2
IMDB-10	40,000	10,000	10

the binary version of IMDB as well as its ten-class version. IMDB-10 is a large movie review dataset that consists of 10 classes [10] and contains 50,000 reviews. IMDB-2 contains 25,000 reviews, and its class is binary. They include movie reviews from around the world, Table 2 show the information of them.

4.2. Experiment settings

Our preprocessing is as follows. For all posts, we use the Natural Language Toolkit (nltk) to obtain the tokens, we divide per subdataset into training and test sets with proportion 9/1. The default parameters in word2vec with the Skip-gram algorithm are used to represent the text in the posts. For the choice of hyper-parameters to train our model, the vector size of the word embedding is 300. The number of filter in CNN-1 layer is 128, in CNN-2 layer is 64, whose size of each filter are both set as 5. And the number of computational units in the LSTM layer set as 100. The unit number of full-connect layer is $H = 100$. We use RMSprop [42] as an optimizer to minimize the objective function and its learning rate α is set to 0.001. Cross-entropy loss function is adopted for training model in this paper. The three subdatasets, IMDB-2 and IMDB-10 are both adopted the same parameter settings. Our network is trained on one NVIDIA GeForce GT730 GPU in a 64-bit Dell computer with one 3.60-GHz CPU and 8GB main memory.

We use accuracy (Acc), precision (P) and recall (R) as metrics (in %) to evaluate the performance of our approach compares against with other models. In addition, since our data is imbalanced, F_1 score is also adopted as evaluation indicator (in %) in this paper. The formula for computing F_1 score is as follow:

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{13}$$

4.3. Comparison methods

The aim of this research is to facilitate instructor teams in MOOCs, more specifically, assist them understand the feeling for their courses in MOOCs of learners in a more efficient way so that they can adjust the content of courses to improve the engagement and satisfaction of learners. Subsequently, how to demonstrate PNNs is an excellent model to reliably identify sentiment of posts? Here, we introduce several existing algorithms, which are widely used to text classification tasks in other NLP fields, as benchmarks to evaluate the effectiveness of our approach. Table 3 shows the comparison methods are used by us.

1. **Bag-of-Words+LR/SVM**
   Wang et al. [44] build an SVM classifier after representing a document with Bag-of-Words (BoW) model. These baselines mainly use machine learning algorithms with BoW as features. We train logistic regression (LR) and SVM respectively with BoW as features.
2. **CNN**
   CNN has strong adaptability and is adept at extracting local features [24]. Here, we also select it for comparison.
3. **LSTM**
   LSTM is a recurrent neural network with memory cells and a three-gate mechanism [14]. It can capture further contextual information than Recurrent Neural Network (RNN) [12] during training. LSTM prevent the gradient vanishing of RNN.
4. **2-layer LSTM**
   There are two hidden layers in 2-layer LSTM structure, both of its two hidden layers are LSTMs. The result of the first hidden layer is sent to the second layer in the same time step, and the function of the second hidden layer is used to capture longer-term dependencies of the input sequence [40].
5. **CNN+2-layer LSTM**
   Huang et al. propose a model based on a CNN and two LSTMs, which is a serial network structure. They employ CNN to obtain useful local features of the text, then features are sent to 2-layer LSTM. Here, the authors' idea is to let two LSTMs extract context-dependent features [16].

Table 3
Comparison methods in this paper

Types	Methods name
Traditional machine learning methods	Logistic regression
	SVM
Neural network methods	CNN
	LSTM
	2-layer LSTM
	CNN + 2-layer LSTM
	2-layer CNN + LSTM
	Self-attention
	textRCNN
	DPCNN

6. **2-layer CNN+LSTM**
   This architecture is similar with our model, but the 2-layer CNN and LSTM are stacked together in a serial manner. We use this model to observe the difference between the NNs combined in a serial manner and in a parallel manner.
7. **Self-attention**
   The Self-attention mechanism is widely used in text classification task of NLP, which provides a more flexible way to select features by increasing their weight according to the importance of features [43]. Moreover, it can capture global information with less computation.
8. **textRCNN**
   Lai et al. use a recurrent structure to capture the semantics of contexts and combine it with a word to present a word [22].
9. **DPCNN**
   DPCNN [18] is a convolutional neural network based on word-level. By deepening the network, DPCNN can extract long distance text dependencies. DPCNN achieves higher accuracy with a small increase in computing cost.

4.4. Results and discussion

The goal of this study is correctly classifying sentiment polarity of posts to facilitate instructors in MOOCs know the feeling of learners. The higher values of metrics for a model present a model has a better performance and more reliable for being applied to sentiment classification in MOOCs posts. In this section, we show the results of PNNs and other comparison methods on three subdatasets.

Table 4 shows the results of PNNs and traditional machine learning algorithms on three subdatasets. These traditional machine learning methods both adopt BoW model to represent feature of sentence. F_1 of PNNs on Humanities, Medicine, and Education is 96.51%, 94.24%, and 92.30%, respectively, which are all higher than the traditional benchmarks. Though BoW+SVM shows best results in traditional methods, our F_1 score exceeds it by 2.68%, 3.21% and 3.46% respectively on three subdatasets. As the results shown in Table 4, PNNs consistently outperforms the traditional approaches on three subdatasets. These demonstrate that PNNs model can avoid to suffer from the data sparsity problem and capture more contextual information of features compared with machine learning using BoW model to represent sentences. Hence, F_1 score of our method has apparently improved. Moreover, the neural network models need not human-designed features at all comparing with traditional approaches that increases sentences sentiment analysis efficiency of us.

Tables 5 and 6 display the results of neural network methods on MOOCs dataset (three subdatasets) and two popular sentiment datasets respectively. As the Table 6 shown, PNNs achieves best results and its accuracies of PNNs are 91.29% and 49.20% on the IMDB-2 and IMDB-10 datasets, surpassing DPCNN by 0.74% and 0.72%, respectively. That means PNNs can work well on various sentiment classification datasets, which is an available and effective model for sentiment classification task. By comparing with other excellent models in Table 5, we can find that, although precision and recall of PNNs are not optimal results on all subdatasets, the F_1 score of PNNs has a better performance than others, outperforming those methods by 1.08%, 1.11% and 1.34%, respectively.

When we compare PNNs with LSTM in Table 5, the F_1 score of the former is higher than the latter on all subdatasets, with improvements increasing by 1.08%, 1.46% and 1.36%, respectively. More specifically, PNNs is higher than CNN in F_1 score on Humanities, Medicine and Education, which rise by 1.25%, 2.29% and 2.53%, respectively. It is because that PNNs combines benefits of these two models, we use two-layer CNNs to extract deep semantic features of sentences and a LSTM to capture long-distance dependence information, then integrating them for sentiment classification. Therefore, PNNs model has a better performance than CNN and LSTM that only use one of aforementioned two types feature.

It is well-known that Self-attention can adjust the weight coefficient of per feature during the training according to the importance of the feature, so as to achieve automatically selecting features. Besides, it can obtain structural information of a sentence via aforementioned calculation way. Because of that, it is also used by us and emerge one of crucial compo-

Table 4
Results of PNNs against with traditional machine learning algorithms on MOOCs datasets

Model	Humanities			Medicine			Education		
	P	R	F_1	P	R	F_1	P	R	F_1
BoW+LR	90.31	94.86	92.53	88.21	94.54	91.26	85.91	88.20	87.03
BoW+SVM	91.86	95.90	93.83	89.76	93.71	91.96	89.62	88.08	88.84
PNNs	**93.35**	**99.89**	**96.51**	**92.20**	**98.36**	**95.17**	**96.43**	**88.50**	**92.30**

P: precision, R: recall, F_1: F_1 score. Best results in each group are in bold. The results of three subdatasets are obtained from the published source code by ourselves.

Table 5
Results of PNNs against with neural network methods on MOOCs datasets

Model	Humanities			Medicine			Education		
	P	R	F_1	P	R	F_1	P	R	F_1
CNN	93.19	97.44	95.26	93.08	92.67	92.88	86.98	92.75	89.77
LSTM	93.21	97.77	95.43	93.97	93.45	93.71	88.31	93.73	90.94
2-layer LSTM	93.24	96.77	94.97	**94.46**	92.89	93.67	88.52	92.87	90.64
CNN+2-layer LSTM	93.16	96.99	95.04	93.10	94.45	93.77	90.16	91.27	90.72
2-layer CNN+LSTM	93.48	97.33	95.36	93.22	94.67	93.94	87.34	93.18	90.17
Self-attention	92.68	96.20	94.40	92.94	95.00	93.96	94.65	82.57	88.20
textRCNN	94.11	96.10	95.10	93.66	94.48	93.92	88.04	94.33	90.96
DPCNN	**93.91**	96.21	95.05	91.50	96.78	94.06	83.41	**99.50**	90.75
PNNs	93.35	**99.89**	**96.51**	92.20	**98.36**	**95.17**	**96.43**	88.50	**92.30**

P: precision, R: recall, F_1: F_1 score. Best results in each group are in bold. The results of all baselines on three subdatasets are obtained from their published source code by ourselves.

Table 6
Experimental results of IMDB-2 and IMDB-10 datasets

Model	IMDB-2	IMDB-10
	Acc	Acc
CNN [39]	86.71	42.88
LSTM [39]	86.04	40.30
2-layer LSTM [39]	89.30	42.64
CNN+2-layer LSTM	89.88	47.40
2-layer CNN+LSTM	89.02	47.78
Self-attention	90.29	48.37
textRCNN [39]	88.84	48.16
DPCNN	90.55	48.48
PNNs	**91.29**	**49.20**

Acc: accuracy. Best results in each group are in bold. The results of CNN, LSTM, 2-layer LSTM, and textRCNN on IMDB-2 and IMDB-10 are cited from [39], the results of another methods on these two datasets are obtained from their published source code by ourselves.

nents our model. However, as can be seen the results of Self-attention model in Table 5, F_1 score of it are all lower than PNNs. We believe the main reason is that comparing with Self-attention model our proposed parallel neural network structure can extract deeper semantic feature and long-distance dependence information from post. By contrast, textRCNN and DPCNN use sentence semantic information alone, they both ignore the structural information of sentence and lack feature selection. Those heavily impact the effect of textRCNN and DPCNN so that F_1 score of PNNs is higher than both two methods. PNNs increases by 2.85%, 1.25%, and 1.34%, respectively on three subdatasets comparing with the former, while its improvements is 1.46%, 1.11% and 1.55% respectively comparing with the latter.

4.5. The effect of strategies on performance

To further investigate the effectiveness of each strategy of our method, we conduct an ablation study on PNNs, separating the effect of each strategy on the F_1 score. The results are illustrated in Table 7. When we directly use 2-layer CNN + LSTM model – serial network structure (LSTM directly stacks on 2-layer CNN) and Self-attention mechanism to sentiment classification, the F_1 score decreases by 1.16% on Education compared with PNNs. This demonstrates that features extracted by CNN are indeed worn out when using LSTM to capture long-distance dependence information in serial network structure, while the parallel structure dose be able to alleviate the problem mentioned before and more effectively preserve features.

Recalling the model architecture, we utilize 2-layer CNNs and LSTM to extract deep semantic feature and dependent information of sentence separately. We find that whether we remove semantic features or long-distance dependence information both have a negative impact on our model. More specifically, ablating 2-layer CNNs part, F_1 score of PNNs drops

Table 7
Ablation performance (F_1 score) of PNNs

Model	Humanities		Medicine		Education	
	F_1	∇	F_1	∇	F_1	∇
PNNs	96.51	–	95.17	–	92.30	–
2-layer CNN+LSTM with Self-attention	95.74	–0.77	94.22	–0.95	91.14	–1.16
PNNs without 2-layer CNN (removing semantic feature)	95.67	–0.84	94.71	–0.46	91.61	–0.69
PNNs without LSTM (removing long-distance dependence information)	95.71	–0.8	94.35	–0.82	91.29	–1.01
PNNs without information fusion	96.00	–0.51	94.84	–0.33	91.69	–0.61
PNNs without Self-attention	96.16	–0.35	94.23	–0.94	90.37	–1.93

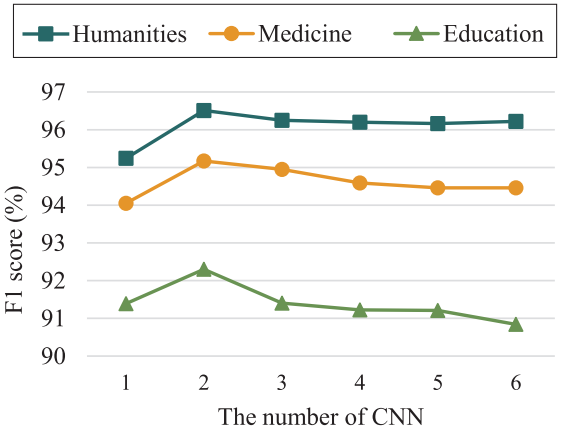

Fig. 3. The number of stack CNN layer.

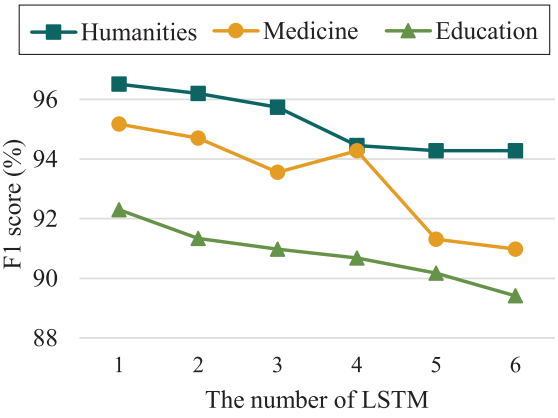

Fig. 4. The number of stack LSTM layer.

by 0.46% – 0.84% on all subdatasets; it evidently falls by 0.8% – 1.01% on same datasets, when we eliminate LSTM. This proves that semantic features and dependent information are both important for classification and helpful to enhance the result of our model.

After that, when we do not fuse semantic feature and long-distance dependence information via full-connect layer, F_1 score of our model declines by 0.61% on Education. It means that the fusion information is more complete representation of a sentence than the independent ones. Although the above strategies provide outstanding contributes to improve effect of PNNs, removing Self-attention mechanism still enables F_1 score of PNNs decrease by 1.93% on Education. Self-attention mechanism is capable of focusing on the useful features at whole contextual semantic features. In conclusion, both of them are indispensable for PNNs to achieve better performance in sentiment classification.

Finally, we evaluate the impact of PNNs with varying parallel neural network structure on three subdatasets, where we achieve this experiment via changing the number of CNN layer or LSTM layer in parallel neural network structure. Figures 3 and 4 indicate the F_1 score of PNNs with different number of stack CNN layers and LSTM layers respectively. We can conclude that the performance becomes better with the increasing of the number of stack layers from Fig. 3. However, with the number of stack layer increasing gradually in Figs. 3 and 4, the F_1 score of PNNs on all subdatasets presents a decreasing trend. To be specifically, the trainable parameters number of PNNs will increases as stack layer rises, while a large number of parameters may cause the model hard to optimized, and even worse the gradient might be exploded or vanished. Therefore, we select 2-layer CNN and a LSTM in parallel neural network structure.

4.6. Analysis of the case study

Different approaches for sentiment classification were compared in previous sections. However, we cannot intuitively feel why a model has a better performance on classification is beneficial for instructors to understand the emotion of learners and promote the course improvement. Because of that, we will list some posts from dataset and show the results of using PNNs model to identify the sentiment polarity of them in this section (shown in Table 8). For compar-

Table 8
Examples of MOOCs discussion forums posts and showing its sentiment polarity classified by PNNs and SVM model

Post example	Score	Sentiment	PNNs	SVM
Terrible interface design! Just put an obvious 'next' button at the bottom of the main body area or clone the whole linear navigation from the top.	1	N	N	N
The Peer Review module is not fully set up yet. You haven't done anything wrong. Professor Boaler simply has a bit more work to do before it is fully ready for us to participate.	3.5	N	N	P
You have to have a higher level of the math that what you are teaching so when a student comes up with a strategy you can help them explain why it works, or doesn't work in all situations.	3.5	N	N	P
Wow! What an exciting group to work and learn with!	7	P	P	P
You are amazing! What a fantastic idea to engage relevent learning and to offer a meaningful experience for students!	7	P	P	P

N: Negative sentiment, P: positive sentiment. "Score" is the original score of the post example.

ison, we also list their classification results obtained by SVM model, because the SVM model shows the best performance on sentiment classification among traditional machine learning methods [28].

In the Table 8, the column of "Score" is the original score of the post, the column of "*Sentiment*" refers to the true sentiment polarity of the post example, and the rest of two columns corresponding to *PNNs* and *SVM* are the classification result of PNNs and SVM models respectively on the post. We find that the posts, which are useful for MOOCs improvements, are mostly negative. Learners will express their dissatisfaction with one aspect of a course in the discussion forum, this dissatisfaction is the cause of their negative emotions or even dropout of the course. Therefore, the instructors should pay more attention on these type of post to know the reasons for the learners' dissatisfaction with the course, so as to timely intervention to support learning and may prevent the learners from giving up the course early.

However, some posts are negative and hold views of the course, which are identified as positive examples by SVM model in Table 8. For instance, in the example of "*The Peer Review module is not fully set up yet. You haven't done anything wrong. Professor Boaler simply has a bit more work to do before it is fully ready for us to participate*", the learner who posted the post clearly stated that the peer review module function is not perfect and the teacher preparation for the course is insufficient. It is not conducive to that the instructors work towards improve MOOCs, because instructors may ignore these negative posts which are classified positive sentiment incorrectly. By contrast, since PNNs have better classification performance than the SVM model, it can classify these posts correctly. So PNNs ensures that the instructors can avoid missing the posts, which are mentioned above, to improve MOOCs effectively.

5. Conclusion

In this paper, we propose a parallel neural network structure with Self-attention mechanism (PNNs) approach for sentiment classification of MOOCs, a novel neural network compound mode which enables PNNs have a better ability of reserving features effectiveness than others. To the best of our knowledge, this is the first study that neural network has been applied to MOOCs sentiment classification domain. In our network, deep semantic feature and long-distance dependence information are captured by parallel neural network structure and merged to obtain contextual semantic representation. Then it is sent to Self-attention mechanism to pay close attention to the useful features and assign more weight on them. Extensive experiments on three MOOCs subdatasets demonstrate the superiority of our proposed model.

In addition, this study provides an effective method to automatically discover the sentiment of posting learners in real time so that it has a potential impact in maximizing instructors' efficiency to monitor MOOCs discussion forums. In the future, we will focus on multilevel classification of MOOCs to achieve more fine-grained sentiment classification in this research field. In this way, instructors can obtain more detailed learners' emotions to help them better analyze learners' learning status and satisfaction with MOOCs, which contributes to optimize course resources to focus on improving the learning experience of learners.

Acknowledgments

The authors would like to thank the anonymous reviewers for the constructive comments. This work was supported in part by the National Natural Sci-

ence Foundation projects of China (No.61877050), in part by the Major research and development projects in Shaanxi Province of China (No. 2019ZDLGY03-10), in part by the Open Project Fund of Shaanxi Province Key Lab. of Satellite and Terrestrial Network Tech of China, and in part by the Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shaanxi Province of China (No. 202160002).

References

[1] J. Mackness and S.F.J. Mak and R. Williams, The ideals and reality of participating in a MOOC. In *Proceedings of the 7 th International Conference on Networked Learning*, (2010), pp. 266–274.

[2] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, Sentiment analysis of Twitter data. In *The Workshop on Languages in Social Media*, (2011), pp. 30–38.

[3] C. Alario-Hoyos, M. Pérez-Sanagustín, C.D. Kloos, Hugo A. Parada G, M. Muñoz Organero and A. Rodríguez-de-las Heras, Analysing the Impact of Built-In and External Social Tools in a MOOC on Educational Technologies. In Davinia HernÃąndez Leo, Tobias Ley, Ralf Klamma, and Andreas Harrer, editors, *Scaling Up Learning for Sustained Impact – 8th European Conference, volume 8095 of Lecture Notes in Computer Science*, (2013), pp. 5–18. Springer.

[4] O. Almatrafi, A. Johri and H. Rangwala, Needle in a haystack: Identifying learner posts that require urgent response in MOOC discussion forums, *Computers & Education* **118** (2018), 1–9.

[5] A. Bakharia, Towards Cross-domain MOOC Forum Post Classification. In Jeff Haywood, Vincent Aleven, Judy Kay, and Ido Roll, editors, *Proceedings of the Third ACM Conference on Learning @ Scale, L@S 2016, Edinburgh, Scotland, UK, April 25 – 26*, (2016), pp. 253–256. ACM.

[6] Y. Bengio, A. Courville and P. Vincent, Representation learning: a review and new perspectives, *IEEE Transactions on Pattern Analysis & Machine Intelligence* **35**(8) (2013), 1798–1828.

[7] Y. Bengio, H. Schwenk, JeansÃl'bastien SenÃl'cal, FrÃl'deric Morin and J. Luc Gauvain, Neural Probabilistic Language Models, *Journal of Machine Learning Research* **3**(6) (2003), 1137–1155.

[8] H.H. Binali, C. Wu and V. Potdar, A new significant area: Emotion detection in E-learning using opinion mining techniques. In *2009 3rd IEEE International Conference on Digital Ecosystems and Technologies*, (2009), pp. 259–264, June 2009.

[9] D. Chaplot, E. Rhim and J. Kim, Predicting Student Attrition in MOOCs using Sentiment Analysis and Neural Networks. volume 1432, June 2015.

[10] Q. Diao, M. Qiu, C.-Y. Wu, A.J. Smola, J. Jiang and C. Wang, Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ' 14*, New York, NY, USA – August 24 – 27, (2014), pp. 193–202.

[11] A. El-Halees, Mining Opinions in User-Generated Contents to Improve Course Evaluation. In Jasni Mohamad Zain, Wan Maseri bt Wan Mohd, and Eyas El-Qawasmeh, editors, Software Engineering and Computer Systems, Communications in Computer and Information Science, (2011), pp. 107–115. Springer Berlin Heidelberg.

[12] J.L. Elman, Finding structure in time, *Cognitive Science* **14**(2) (1990), 179–211.

[13] B. Handal, J. Mac Nish and P. Petocz, Academics adopting mobile devices: The zone of free movement, *30th Annual conference on Australian Society for Computers in Learning in Tertiary Education, ASCILITE 2013*, pp. 350–361, January 2013.

[14] S. Hochreiter and J. Schmidhuber, Long Short-Term Memory, *Neural Computation* **9**(8) (1997), 1735–1780.

[15] E.H. Huang, R. Socher, C.D. Manning and A.Y. Ng, Improving word representations via global context and multiple word prototypes. In *Meeting of the Association for Computational Linguistics: Long Papers*, (2012), pp. 873–882.

[16] Q. Huang, R. Chen, X. Zheng and Z. Dong, Deep Sentiment Representation Based on CNN and LSTM, In *2017 International Conference on Green Informatics (ICGI)*, (2017), pp. 30–33, August 2017.

[17] R. Johnson and T. Zhang, Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar, editors, NAACL HLT 2015, *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, USA, May 31 – June 5, 2015, pp. 103–112. The Association for Computational Linguistics, 2015.

[18] R. Johnson and T. Zhang, Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, Vancouver, Canada, July 30 – August 4, Volume 1: Long Papers, (2017), pp. 562–570.

[19] E. Jung, D. Kim, M. Yoon, S. Park and B. Oakley, The influence of instructional design on learner control, sense of achievement, and perceived effectiveness in a supersize MOOC course, *Computers & Education* **128** (2019), 377–388.

[20] N. Kalchbrenner, E. Grefenstette and P. Blunsom, A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers, (2014), pp. 655–665. The Association for Computer Linguistics.

[21] Y. Kim, Convolutional Neural Networks for Sentence Classification. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, (2014), pp. 1746–1751. ACL, 2014.

[22] S. Lai, L. Xu, K. Liu and J. Zhao, Recurrent Convolutional Neural Networks for Text Classification. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, (2015), pp. 2267–2273. AAAI Press.

[23] J. Li, T. Luong, D. Jurafsky and E.H. Hovy, When Are Tree Structures Necessary for Deep Learning of Representations? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, Lisbon, Portugal, September 17-21, (2015), pp. 2304–2314. The Association for Computational Linguistics, 2015.

[24] Y. LÃl'cun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE* **86**(11) (1998), 2278–2324.

[25] S. Mak and R. Williams, Blogs and forums as communication and learning tools in a MOOC. In *International Conference on Networked Learning (NLC '10)*, (2010), pp. 275–285.

[26] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality. In *International Conference on Neural Information Processing Systems*, (2013), pp. 3111–3119.

[27] A. Mnih and G. Hinton, Three new graphical models for statistical language modelling. In *International Conference on Machine Learning*, (2007), pp. 641–648.

[28] P.M. Moreno-Marcos, C. Alario-Hoyos, P.J. MuÃ´soz Merino, I. EstÃl'vez-Ayres, and C.D. Kloos, Sentiment analysis in MOOCs: A case study. In *2018 IEEE Global Engineering Education Conference, EDUCON 2018, Santa Cruz de Tenerife, Tenerife, Islas Canarias, Spain, April 17-20*, (2018), pp. 1489–1496. IEEE, 2018.

[29] J. Moshinskie, How To Keep E-Learners from E-Scaping, *Performance Improvement* **40**(6) (2001), 28–35.

[30] T. Phan, S.G. Mcneil and B.R. Robin, Students' patterns of engagement and course performance in a Massive Open Online Course, *Computers & Education* **95** (2016), 36–44.

[31] R. Piryani, M. Devaraj and V.K. Singh, Analytical mapping of opinion mining and sentiment analysis research during 2000-2015, *Inf Process Manage* **53**(1) (2017), 122–150.

[32] W.W. Porter, C.R. Graham, K.A. Spring and K.R. Welch, Blended learning in higher education: Institutional adoption and implementation, *Computers & Education* **75** (2014), 185–195.

[33] A. Ramesh, S.H. Kumar, J. Foulds and L. Getoor, Weakly Supervised Models of Aspect-Sentiment for Online Course Discussion Forums. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (2015), pp. 74–83.

[34] G. Rao, W. Huang and Z. Feng, LSTM with sentence representations for document-level sentiment classification, *Neurocomputing* (2018), pp. 49–57.

[35] C. Osvaldo Rodriguez, MOOCs and the AI-Stanford Like Courses: Two Successful and Distinct Course Formats for Massive Open Online Courses, *European Journal of Open, Distance and E-Learning*, 2012.

[36] D. Shah, By The Numbers: MOOCS in 2017. January 2018.

[37] T. Sinha and J. Cassell, Connecting the Dots: Predicting Student Grade Sequences from Bursty MOOC Interactions over Time. In Gregor Kiczales, Daniel M. Russell, and Beverly Park Woolf, editors, Proceedings of the Second ACM Conference on Learning @ Scale, L@S 2015, Vancouver, BC, Canada, March 14 – 18, (2015), pp. 249–252. ACM, 2015.

[38] D. Song, H. Lin and Z. Yang, Opinion Mining in e-Learning System. In *2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007)*, (2007), pp. 788–792, September 2007.

[39] X. Sun, Y. Gao, R. Sutcliffe, S. Guo, X. Wang and J. Feng, Word representation learning based on bidirectional grus with drop loss for sentiment classification, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, (2019), pp. 1–11.

[40] K.S. Tai, R. Socher and C.D. Manning, Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks, *Computer Science* **5**(1) (2015), 36.

[41] D. Tang, B. Qin and T. Liu, Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *Conference on Empirical Methods in Natural Language Processing*, (2015), pp. 1422–1432.

[42] Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, Attention is All you Need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, (2017), pp. 6000–6010.

[44] S. Wang and C.D. Manning, Baselines and bigrams: simple, good sentiment and topic classification. In *Meeting of the Association for Computational Linguistics: Short Papers. Association for Computational Linguistics*, volume 2, (2012), pp. 90–94. The Association for Computer Linguistics, 2012.

[45] M. Wen, D. Yang and C. Penstein RosÃl'. Sentiment Analysis in MOOC Discussion Forums: What does it tell us? In John C. Stamper, Zachary A. Pardos, Manolis Mavrikis, and Bruce M. McLaren, editors, Proceedings of the 7th International Conference on Educational Data Mining, EDM 2014, London, UK, July 4-7, (2014), pp. 130–137. International Educational Data Mining Society (IEDMS).

[46] T. Wilson, J. Wiebe and P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, *Proceedings of HLT/EMNLP-05* **7**(5) (2005), 347–354.

[47] X. Wu, Y. Cai, Q. Li, J. Xu and H-f. Leung, Combining contextual information by self-attention mechanism in convolutional neural networks for text classification. In Web Information Systems Engineering – WISE 2018 – 19th International Conference, Dubai, United Arab Emirates, November 12-15, 2018, Proceedings, Part I, (2018), pp. 453–467.

[48] J. Xu, D. Chen, X. Qiu and X. Huang, Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification. In Jian Su, Xavier Carreras, and Kevin Duh, editors, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, (2016), pp. 1660–1669. The Association for Computational Linguistics, 2016.

[49] H. Zhang, J. Wang, J. Zhang and X. Zhang, YNU-HPCC at SemEval 2017 Task 4: Using A Multi-Channel CNN-LSTM Model for Sentiment Classification. In Steven Bethard, Marine Carpuat, Marianna Apidianaki, Saif M. Mohammad, Daniel M. Cer, and David Jurgens, editors, Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3–4, (2017), pp. 796–801. Association for Computational Linguistics, 2017.