Online-Journals.org

iJET International Journal of Emerging Technologies in Learning

iJET | eISSN: 1863-0383 | Vol. 18 No. 15 (2023) | OPEN ACCESS

https://doi.org/10.3991/ijet.v18i15.42383

PAPER

Students' Composition Evaluation Model Based on a Natural Language Processing Algorithm

Lin Wang(✉), **Weifeng Deng**

Hainan Vocational University of Science and Technology, Haikou, China

wanglintaichi@163.com

ABSTRACT

It is subjective, time consuming and labor intensive to evaluate students' compositions. Use of natural language processing (NLP) technology effectively improves the evaluation efficiency and reduces the burden on teachers. In order to overcome the problems of traditional models, such as over-fitting and poor generalization ability, this research studied a students' composition evaluation model based on an NLP algorithm. A students' composition evaluation model based on a multi-task learning framework was proposed, which completed three sub-tasks simultaneously using the NLP algorithm. Three different encoding methods were used; namely, convolutional neural network (CNN), recurrent neural network (RNN), and long short-term memory (LSTM), which captured text information from multiple perspectives. A new pairing pre-training mode was built, which aimed to help build an NLP-based students' composition evaluation model based on the multi-task learning framework, thus alleviating the deviation caused by excessive correlation. The experimental results verified that the constructed model and the proposed method were effective.

KEYWORDS

natural language processing, students' composition evaluation, multi-task learning framework

1 INTRODUCTION

With the rapid development of information technology, the application of artificial intelligence in various fields is becoming increasingly widespread. As an important branch of artificial intelligence, natural language processing (NLP) has received extensive attention in several fields, such as linguistics, computer science, and information engineering [1–5]. In recent years, the research of NLP-based students' composition evaluation models has become a hot topic. Use of NLP technology effectively improves the evaluation efficiency and reduces the burden on teachers, because it is subjective, time consuming, and labor intensive to evaluate the compositions of students [6–11].

Wang, L., Deng, W. (2023). Students' Composition Evaluation Model Based on a Natural Language Processing Algorithm. *International Journal of Emerging Technologies in Learning (iJET)*, 18(15), pp. 52–66. https://doi.org/10.3991/ijet.v18i15.42383

Article submitted 2023-04-13. Resubmitted 2023-06-06. Final acceptance 2023-06-17. Final version published as submitted by the authors.

The research on students' composition evaluation models began in the 1980s, and early studies were mainly based on artificial rules and statistical methods. With the development of deep learning technology, significant breakthroughs have been made in the NLP field. In particular, the proposal of a bidirectional encoder representations from transformers (BERT) model has brought revolutionary progress to NLP tasks [12–15]. Performance has been significantly improved in various NLP tasks using BERT-based pre-training models, such as generative pre-trained transformer (GPT) series of models [16–18].

In the study of students' composition evaluation models, the traditional method based on feature engineering has been gradually replaced with the deep learning–based method. In recent years, researchers have tried to evaluate students' compositions using several neural network structures, such as CNN, RNN, LSTM, and Transformer [19–22], which have achieved good results in students' composition-evaluation tasks. However, they still have certain limitations, such as over-fitting and poor generalization ability.

In order to overcome these problems, this research studied the students' composition evaluation model based on an NLP algorithm by taking English teaching as an example. Combined with the latest pre-training model and neural network structures, the ways to improve the model performance were discussed. A students' composition evaluation model based on the multi-task learning framework is proposed (Section 2), which completed three sub-tasks simultaneously using an NLP algorithm. Three encoding methods o(CNN, RNN, and LSTM) are used, which captured text information from multiple perspectives. A new pairing pre-training mode was built (Section 3), which aimed to help build an NLP-based students' composition evaluation model based on the multi-task learning framework, thus alleviating the deviation caused by excessive correlation. The experimental results verify that the constructed model and the proposed method were effective.

2 CONSTRUCTION OF COMPOSITION EVALUATION MODEL STRUCTURE

A students' composition evaluation model based on a multi-task learning framework was proposed in this study, which completed three sub-tasks simultaneously using the NLP algorithm. These sub-tasks shared the information-coding part, including the input layer, the embedding layer, and the semantic encoding layer. In the semantic encoding layer, CNN, RNN, and LSTM were used to construct the model, respectively. The main advantage was that different tasks shared the same input, embedding, and semantic encoding layers in the multi-task learning framework, meaning that they shared the same parameters, which reduced the number of model parameters and the computational complexity of the training model, thus improving the training efficiency. Figure 1 shows the network structure diagram of the model.

One important advantage of multi-task learning is knowledge transfer between different tasks. In the students' composition evaluation model in this study, the information coding part was shared, which enabled the models with different structures to share the extracted semantic information, such as CNN, RNN, and LSTM, thus helping improve the generalization ability and performance of the model in various sub-tasks.

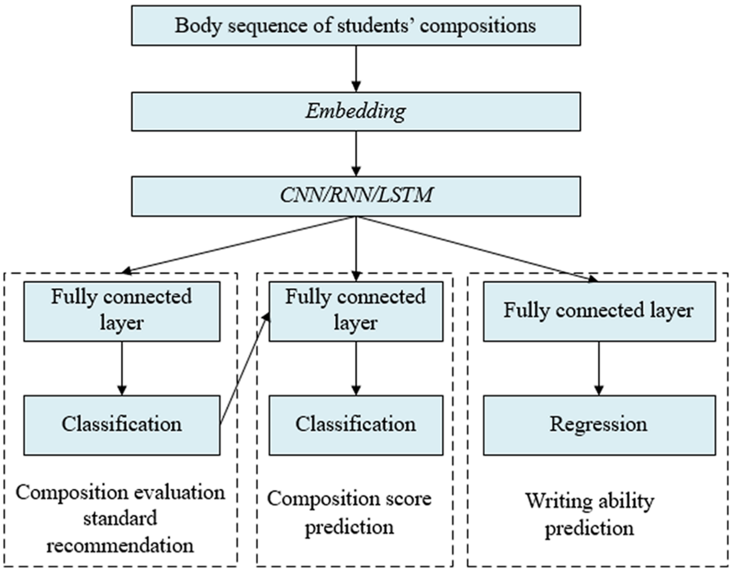
Body sequence of students' compositions
Embedding
CNN/RNN/LSTM
Fully connected layer
Classification
Composition evaluation standard recommendation
Fully connected layer
Classification
Composition score prediction
Fully connected layer
Regression
Writing ability prediction

Fig. 1. Network structure diagram of the model

Specifically, the CNN-based encoding layer had great potential in NLP tasks and showed a strong performance especially in text classification and sentiment analysis tasks. In NLP tasks, the convolutional layer was used to recognize local semantic features in text, such as n-gram, words and phrases, which was very important to understand the text semantics and evaluate the composition quality. At the same time, the convolution operation had translation invariance, meaning that the convolutional layer detected the same features at different positions of input data. In text-processing tasks, this meant that the model recognized key information in the text, regardless of where it appeared in the text.

After obtaining the output through the embedding layer, the body of students' compositions was represented as a word sequence in the CNN-based encoding layer. Let z_u be each word, f be the embedded dimension of each word, and b be the length of the composition body sequence. Then there were:

$$z_{u:b} = z_1 \oplus z_2 \oplus \cdots \oplus z_b \tag{1}$$

If l convolutions were set in the convolutional layer, and the height g of the convolution kernel was equal to the number of words taken in each sliding window, then the convolution kernel satisfied $Q \in R^{l\times(g\times f)}$. Let v_u be the result feature of the sliding window, $z_{u:u+g-1}$ be the word embedding matrix in the u-th window, $n \in R$ be the bias vector, and d be the nonlinear function. The calculation results of the convolutional layer were obtained by the following equation:

$$v_u = d(Q \cdot z_{u:u+g-1}) + n \tag{2}$$

Let b be the length of the composition body sequence, g be the height of the convolution kernel, $b - g + m$ be the number of window slides, and $v = \{v_1, v_2, \ldots, v_{b-g+1}\}$ be the results of the convolved convolution kernel. The output results of the convolutional layer were further maximally pooled, i.e., $\hat{v} = MAX\{v\}$. The ultimately pooled data was represented by $g = \{\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_l\}$, because l convolution kernels existed.

The RNN-based encoding layer was widely used in NLP tasks and showed a superior performance, especially in processing text data with sequence structure. RNN naturally handled input sequences with variable length, which was very suitable for processing text data with different lengths, thus allowing the model to flexibly handle compositions with different lengths, to improve the evaluation effectiveness. At the same time, RNN was naturally suitable for processing sequence data, because its structure captured sequence information through loop join, which helped the model capture important information, such as grammatical structure and tense relationships in the text in students’ composition evaluation tasks, thus better evaluating the composition quality.

Let the y-th word z_y in the composition body sequence be the RNN input at time y, a_y be the value of the hidden layer, and g_y be the output of the encoding layer. It is worth noting that the size of a_y was determined by both the input z_y and a_{y-1} of the previous time. Therefore, each cell unit of RNN included calculation in two aspects: the hidden layer, and the output vector g_y obtained by combining z_y with a_{y-1}. Let I, Q, and T be the weight vectors, and n and v be the bias vectors. The specific calculation equations are as follows:

$$a_y = \delta(Iz_y + Qa_{y-1} + n) \tag{3}$$

$$g_y = \delta(Ca_y + v) \tag{4}$$

Both the weight and bias vectors were shared in each RNN cell. After the entire body of a student’s composition was processed by an RNN encoder, the output vector g of the last cell was finally output.

The LSTM network more effectively captured the long-term dependencies in the input sequence by introducing a cell state and a gate mechanism. A special activation function and gate mechanism were designed, which transmitted the gradient more stably in the back-propagation process, thus effectively alleviating these two problems.

In the LSTM-based encoding layer, the network “forget” gate determined the information to be forgotten, i.e., discarded from the cell state of the previous time. Let g_{y-1} be the output of the previous state and z_y be the input information of the current state. After g_{y-1} and z_y were processed by a sigmoid function, the processed results were multiplied by the retained cell state of the previous time, which aimed to determine how much forgotten information was retained. Let Q_d be the weight vector of z_y, I_d be the weight vector of g_{y-1} output by the hidden layer of the previous time, and n_d be the bias vector. Then the calculation equation of the forget gate d_y is given as follows:

$$d_y = \delta(Q_d z_y + I_d g_{y-1} + n_d) \tag{5}$$

The input gate determined the information to be retained at the current time. Let Q be the weight vector of z_y, I_u be the weight vector of g_{y-1} output by the hidden layer of the previous time, and n_u be the bias vector. The output of the input gate was obtained using g_{y-1} and z_y and processed by a sigmoid function. The specific equation was as follows:

$$u_y = \delta(Q_u z_y + I_u g_{y-1} + n_u) \tag{6}$$

Candidate new information $\tilde{V}_y$ was generated using g_{y-1} and z_y and processed by tanh activation function:

$$\tilde{V}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{7}$$

The information to be forgotten was combined with the candidate new information, which obtained the new state v_y at this time:

$$v_y = d_y \circ v_{y-1} + u_y \circ \tilde{V}_y \tag{8}$$

Let Q_p be the weight vector of the input ${}_tz_y$ at the current time, g_{y-1} be the hidden-layer output of the previous time, I_p be the corresponding weight vector, and n_p be the bias vector. The output gate equation is given as follows:

$$p_y = \delta(Q_p z_y + I_p g_{y-1} + n_p) \tag{9}$$

The output of the output gate was combined with v_y processed by tanh, which obtained the output information g_y of the LSTM cell:

$$g_y = p_y \circ TAg(v_y) \tag{10}$$

Text information was captured from multiple perspectives using three encoding methods; namely, CNN, RNN, and LSTM. CNN captured local syntactic and semantic features, and RNN and LSTM captured long-distance dependencies and sequence features. This multi-model fusion strategy helped improve the generalization ability of the model, enabling it to better capture features in different types of texts. Therefore, the input word vectors were encoded using CNN, RNN, and LSTM, respectively, and the results recommended by composition evaluation standards were incorporated into the score prediction tasks, after considering the impact of the standards on scores. This approach enabled the model to better focus on evaluation dimensions related to scores, such as content, structure, and grammar, which not only helped improve the accuracy of the mode in composition quality evaluation but also made the predicted results of the model more in line with human evaluation standards. Figure 2 shows the flowchart of scoring students' compositions.

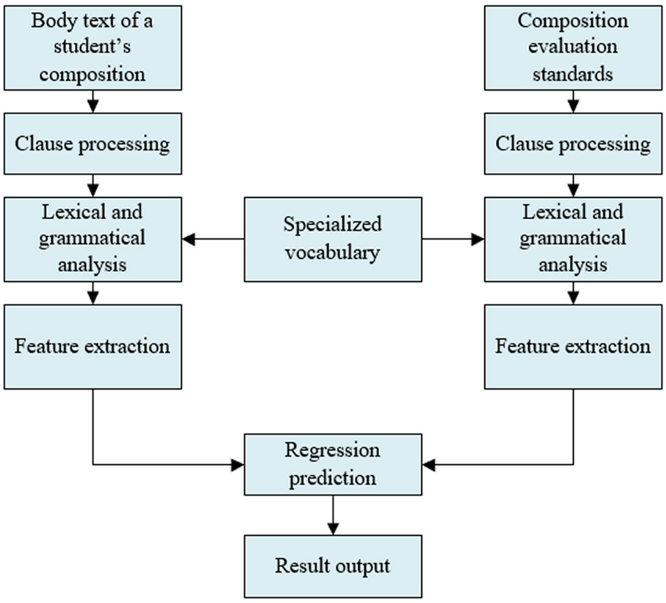
Body text of a student's composition
Composition evaluation standards
Clause processing
Clause processing
Lexical and grammatical analysis
Specialized vocabulary
Lexical and grammatical analysis
Feature extraction
Feature extraction
Regression prediction
Result output

Fig. 2. Students' english composition scoring flowchart

Therefore, for the sub-tasks of composition evaluation standard recommendation y_1 and composition score prediction y_2, softmax was used in this study to obtain classification results in the final stage. Let Q_k^o be the weight vector of y_k, n_k^o be the bias vector of y_k, $y_k \in \{y_1, y_2\}$ be the prediction task, Q_3^o be the weight vector of y_3, and n_3^o be the bias vector of y_3. Then there is:

$$\hat{t}_k = SM(Q_k^o g_k + n_k^o) \tag{11}$$

The writing ability prediction y_3 was processed using the RELU function, which obtained the regression prediction results:

$$\hat{t}_3 = RELU(Q_3^o g_3 + n_3^o) \tag{12}$$

The gap between the prediction result $\hat{t}$ and the true value t was minimized. The cross entropy loss function was used for the sub-tasks of composition evaluation standard recommendation y_1 and composition score prediction y_2.

$$M_k(\hat{t}_k, t_k) = -\sum_{j=1}^{|t_k|} t_{k,j} \log(\hat{t}_3 - t_3)^2 \tag{13}$$

For the sub-task of writing ability prediction y_3, standard deviation was used as the loss function:

$$M_3(\hat{t}_3, t_3) = \frac{1}{b}\sum_{u=1}^{b} (\hat{t}_3 - t_3)^2 \tag{14}$$

Let η_k be the weight of each prediction task. Then the total loss function is given by the following equation:

$$M = \sum_{k=1}^{|Y|} \eta_k M_k(\hat{t}_k, t_k) \tag{15}$$

3 PAIRING THE PRE-TRAINING MODE OF THE EVALUATION MODEL

Different tasks may have excessive correlation in a multi-task learning framework, which can make the model focus on certain features while ignoring other important information. Therefore, a new pairing pre-training mode was constructed in this study, which aimed to help build an NLP-based students' composition evaluation model based on the multi-task learning framework, thus alleviating the deviation caused by excessive association and enabling the model to focus on the correlation between label and comment texts and aspect entity/category words. This approach helped the model better learn the correlation between texts and aspect entity/category words, which improved the performance of the model in students' composition evaluation tasks. Figure 3 shows different samples with the same aspect entity/category words.

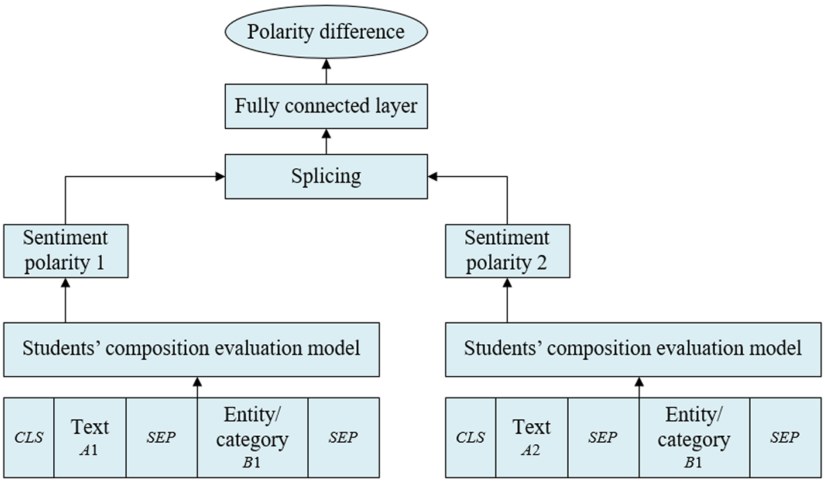

Fig. 3. Different samples with the same aspect entity/category words

The input data was usually the word vector in NLP tasks. Adversarial training was realized by adding certain disturbance to parameters of the word-embedding layer. Specifically, the training aimed to find a disturbance, which caused the maximum performance reduction of the model after being added to the original data. This process was regarded as a max-min task, which minimized the loss of the model in the adversarial example and maximized the difference between the original data and the adversarial example.

Adversarial training was introduced into the students' composition evaluation model in this study, which aimed to enable the model to learn the disturbance information in the adversarial example, thus better handling different types of compositions in practical applications, including those that might contain spelling and grammar errors or other "noise."

Let ϕ be the network parameter, e_{AD} be the adversarial disturbance, z be the data feature, t be the data label, and z and t be from a certain distribution F. The size of adversarial disturbance e_{AD} was limited to within r. The expression of adversarial training was given by the following equation:

$$\underset{\varphi}{MIN}\, R_{(z,t)\sim F} \underset{\|e_{AD}\|\le r}{MAX} m(\varphi; z+e_{AD}, t) \tag{16}$$

A free batch adversarial training algorithm was proposed in this study, which was an improved adversarial training method aiming at improving the robustness of neural network models. Unlike traditional adversarial training methods, this algorithm comprehensively utilized the gradients of both disturbances and model parameters when calculating gradients, thus improving the training efficiency. Let β be the hyper-parameter, and h be the step size of disturbance. Then the disturbance term equation of the fast gradient method is given as follows:

$$e_{AD} = \beta * h \big/ \|h\|_2 \tag{17}$$

$$h = \nabla_z M(\varphi, z, t) \tag{18}$$

$$e_{AD}^{y+1} = \prod\nolimits_{\|e_{AD}\|D\le r} \left(e_{AD}^{y} + \beta h(e_{AD}^{y}) \big/ \|h(e_{AD}^{y})\|_2 \right) \tag{19}$$

$$h(e_{AD}^{y}) = \nabla_{e_{AD}^{y}} M(d_{\varphi}(Z + e_{AD}^{y}), t) \tag{20}$$

This algorithm comprehensively utilized the gradients of both disturbances and model parameters, which effectively reduced information waste in gradient calculation, thus helping improve training efficiency and accelerate model convergence. At the same time, the comprehensive utilization of both gradients enabled the model to better learn the difference between the original data and the adversarial example, which helped improve the robustness of the model when dealing with data with noise.

This algorithm was called the free batch adversarial training algorithm, because it calculated the gradients of J adversarial examples ($Z + e_{AD}^{0}$, $Z + e_{AD}^{1}$, ..., $Z + e_{AD}^{J-1}$) in J iterations. Let e_{AD}^{y} be the y-th disturbance, h be the gradient in the y-th backpropagation, and $h(e_{AD}^{y})$ be the gradient generated on the basis of adversarial example. It was assumed that $h(e_{AD}^{y})$ was used to update the disturbance. It was seen that h_y gradually accumulated during the iteration process, and its mean value was used for gradient descent. This algorithm had higher training efficiency and better convergence effect, because it fully utilized the gradient of model parameters in J disturbances.

The free batch adversarial training process included two stages of pre-training and formal training, data processing, and generation during these stages. The basic process of this process included raw data pre-processing, building samples for the pairing pre-training stage and the formal training stage, carrying out pairing pre-training and formal training, and obtaining the final indexes. Three variants were designed based on the basic process; namely, modifying samples, modifying model construction based on the multi-task learning framework, and using adversarial training. After evaluating these variants in internal comparative experiments, the best solution was selected and compared with existing benchmark algorithms. Figure 4 shows the algorithm flowchart.

The above design method had systematic evaluation and was flexible, scientific, and practical, which helped find the optimal solution, thus improving the performance of the students' composition evaluation model.

$$G(o,w) = -\sum_{z}(o(z)LOw(z) + (1-o(z))LO(1-w(z))) \tag{21}$$

$$SM(t_u) = \frac{r^{t_u}}{\sum_{j=1}^{b} r^{t_u}} \tag{22}$$

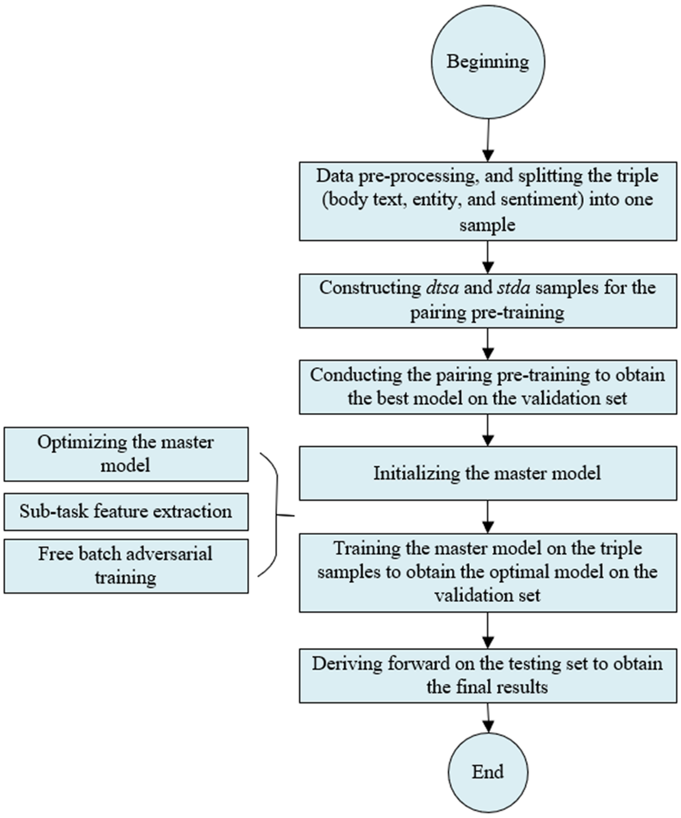
Beginning
Data pre-processing, and splitting the triple (body text, entity, and sentiment) into one sample
Constructing dtsa and stda samples for the pairing pre-training
Conducting the pairing pre-training to obtain the best model on the validation set
Initializing the master model
Optimizing the master model
Sub-task feature extraction
Free batch adversarial training
Training the master model on the triple samples to obtain the optimal model on the validation set
Deriving forward on the testing set to obtain the final results
End

Fig. 4. Algorithm flowchart

4 EXPERIMENTAL RESULTS AND ANALYSIS

Table 1 shows the performance of several single-task and multi-task learning methods in three tasks; namely, composition evaluation standard recommendation, composition score prediction, and writing-ability prediction. Specifically, these methods include the single-task learning methods based on term frequency-inverse document frequency + support vector machine (TF – IDF + SVM), CNN, RNN, and LSTM, as well as the multi-task learning methods based on CNN, RNN, and LSTM.

It can be seen from the table that the performance of multi-task learning methods is generally better than that of single-task learning methods in all tasks, indicating that multi-task learning shares effective feature representations among different tasks, thus improving the performance of each task. In single-task learning methods, RNN and LSTM perform relatively well in each task, maybe because these two methods capture long-distance dependencies in text, thus better understanding and representing text information. Among the multi-task learning methods, the LSTM-based method achieved the highest performance (87.57) in the composition score prediction task, while the CNN-based method achieved the highest performance (76.59) in the writing-ability prediction task, indicating that different multi-task learning methods may have advantages in different tasks. Overall, the experimental results indicated that the multi-task learning framework had better performance in tasks such as composition evaluation standard recommendation, composition score prediction, and writing ability prediction, compared with the single-task learning framework. Therefore, adopting the multi-task learning method proposed in this study for these tasks may be a more effective strategy.

Table 1. Comparison of experimental results

		Composition Evaluation Standard Recommendation	Composition Score Prediction	Writing-Ability Prediction
Single-task learning	TF – IDF + SVM	72.14	75.41	53.62
	CNN	81.62	81.06	65.41
	RNN	86.59	83.52	61.42
	LSTM	80.35	84.95	75.43
Multi-task learning	CNN	84.16	81.26	76.59
	RNN	84.59	83.48	70.15
	LSTM	83.62	87.57	73.48

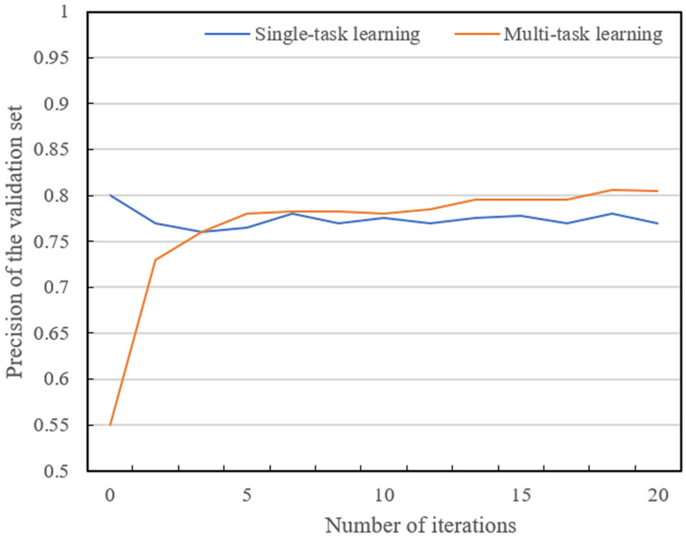

Fig. 5. Performance comparison between single-task and multi-task learning frameworks

Figure 5 shows the performance of single-task and multi-task learning methods with different iterations. Specifically, the horizontal axis represents the number of iterations, and the vertical axis represents performance indexes, such as accuracy or loss value. It can be seen from the figure that the performance of multi-task learning is better than that of single-task learning when the number of iterations is small (0 to 5), maybe because multi-task learning shares effective feature representations among different tasks, thus learning useful information with fewer iterations. As the number of iterations increases, the performance gap between single-task and multi-task learning gradually narrows, indicating that the single-task learning method gradually learns effective feature representations as the training progresses but may require more iterations. When the number of iterations is high (from 15 to 20), the performance of multi-task learning is still slightly better than that of single-task learning, indicating that multi-task learning always has certain advantages throughout the entire training process. The experimental results showed that the multi-task learning framework had better performance than the single-task learning framework in different iterations. Therefore, adopting the multi-task learning method may be a more effective strategy in scenarios where better performance needs to be achieved with fewer iterations. Meanwhile, the multi-task learning consistently exhibited certain advantages throughout the entire training process, which also demonstrated its effectiveness in handling multiple tasks.

Table 2. Comparative experimental results of different models

Models	Accuracy (%)	Recall (%)	Precision (%)	F1 Value (%)
CNN	66.4	83.4	62.5	75.8
RNN	72.9	88.2	74.8	71.6
LSTM	71.8	94.6	70.6	82.9
BERT	85.7	83.7	82.9	81.6
Model in this study	94.2	95.1	85.7	95.3

Table 3. Comparative experimental results of different pooling strategies

Pooling Strategies	Accuracy (%)	Recall (%)	Precision (%)	F1 Value (%)
Maximum pooling	90.5	91.3	97.5	93.7
Average pooling	95.8	91.4	90.3	97.5

Table 4. Comparative experimental results of different models in an ablation experiment

Models	Accuracy (%)	Recall (%)	Precision (%)	F1 Value (%)
Complete model	95.8	91.6	93.8	90.2
Removing CNN encoding layer	91.6	98.5	91.5	96.4
Removing RNN encoding layer	85.7	84.1	83.1	85.7
Removing LSTM encoding layer	82.4	81.6	94.8	92.4

Table 2 shows the performance of several models in classification tasks; namely, CNN, RNN, LSTM, BERT, and the model in this study. The table lists the accuracy, recall, precision, and F1 value of each model. It can be seen from the table that the model in this study exhibits high performance in terms of accuracy, recall, precision, and F1 value, reaching 94.2%, 95.1%, 85.7%, and 95.3%, respectively, indicating that the model has good generalization ability in this classification task. The BERT model also performs relatively well in terms of accuracy, precision, and F1 value, reaching 85.7%, 82.9%, and 81.6%, respectively, but has slightly lower recall than the model in this study, maybe because BERT, as a pre-training model, utilizes a large amount of unlabeled data to learn universal language representations, thus improving the model's performance. Among the three models of CNN, RNN, and LSTM, RNN performs the best in accuracy (72.9%) but is slightly inferior to the other two models in recall, precision, and F1 value, maybe because RNN captures the sequence information of text, but it may perform poorly in dealing with long-distance dependencies. The LSTM model exhibits high performance in terms of recall (94.6%) but has slightly lower precision and F1 value than other models, maybe because LSTM has certain advantages in handling long-distance dependencies, but its performance in other aspects is not balanced enough. The experimental results showed that the model in this study had optimal performance in classification tasks and exhibited good generalization ability, maybe because the model fully utilized the advantages of different models, thus achieving high performance in various performance indexes.

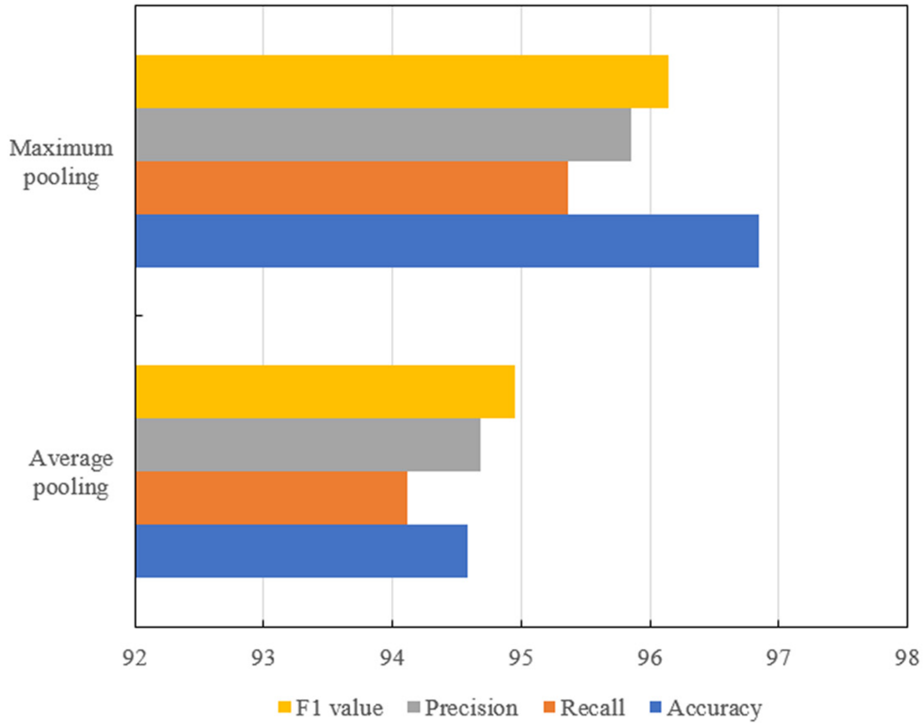

Fig. 6. Experimental results of different pooling strategies

The performance of two different pooling strategies (i.e., maximum and average pooling) in classification tasks can be seen in Table 3 and Figure 6. The accuracy, recall, precision, and F1 value of each pooling strategy are listed. According to the experimental results, the average pooling strategy outperforms the maximum pooling strategy in terms of accuracy and F1 value, reaching 95.8% and 97.5%, respectively, indicating that the average pooling strategy may be more superior in overall performance. The performance of both strategies is quite similar in terms of recall, reaching 91.4% and 91.3%, respectively, indicating that they have a similar effect in correctly identifying positive samples. The performance of the maximum pooling strategy is better in terms of precision, reaching 97.5%, while the precision of the average pooling strategy is 90.3%, indicating that the maximum pooling strategy may have an advantage in predicting the accuracy of positive samples. The experimental results indicated that the average pooling strategy had good overall performance and had advantages, especially in terms of accuracy and F1 value. However, the maximum pooling strategy exhibited better performance in terms of precision. Therefore, appropriate pooling strategies could be selected based on specific task requirements and the importance of performance indexes in practical applications.

Table 4 compare the results of various models in an ablation experiment. Performance of the complete model and the model with different encoding layers removed in the classification tasks in the ablation experiment can be seen in Table 4. The table lists the accuracy, recall, precision, and F1 value of each model. The experimental results show that the complete model performed well in all performance indexes, reaching 95.8%, 91.6%, 93.8%, and 90.2%, respectively, indicating that the model has high generalization ability in classification tasks. The model without the CNN encoding layer performed best in terms of recall, reaching 98.5%, but was slightly inferior to the complete model in terms of accuracy, precision, and F1 value, maybe because the CNN encoding layer captured local features, which had a certain impact on the model's performance. The model without the RNN encoding layer had lower performance in terms of accuracy, recall, precision, and F1 value, maybe because the RNN encoding layer captured the sequence information of the text, making significant contributions to the model's performance. The model without the LSTM encoding layer performed the best in precision, reaching 94.8%, but performed poorly in other performance indexes, maybe because the LSTM encoding layer had advantages in handling long-distance dependencies, but its performance in other aspects was not balanced enough. The experimental results show that the complete model had optimal performance in classification tasks. In various ablation experiments, removing the RNN encoding layer had the greatest impact on the model's performance, while removing the CNN and LSTM encoding layers had a relatively small impact, indicating that the RNN encoding layer in the model in this study had significant contributions to the model's performance, while the CNN and LSTM encoding layers had relatively small contributions. These conclusions provide references for model design and optimization in practical applications.

5 CONCLUSION

This research studied the students' composition evaluation model based on an NLP algorithm. A students' composition evaluation model based on the multi-task learning framework was proposed, which completed three sub-tasks simultaneously using the NLP algorithm. Three encoding methods of CNN, RNN, and LSTM were used to capture text information from multiple perspectives. A new pairing pre-training mode was built, which aimed to help build an NLP-based students' composition evaluation model based on the multi-task learning framework, thus alleviating the deviation caused by excessive correlation. Based on previous experimental results and analysis, the above experimental results are summarized as follows:

(1) In the comparative experiment of different models, the model in this study showed high performance in terms of accuracy, recall, precision, and F1 value, and

demonstrated good generalization ability. The experimental results showed that the model had optimal performance in classification tasks.

(2) In the comparative experiment of different pooling strategies, the average pooling strategy had better overall performance and had more advantages, especially in terms of accuracy and F1 value. However, the maximum pooling strategy exhibited better performance in terms of precision. Therefore, appropriate pooling strategies could be selected based on specific task requirements and the importance of performance indexes in practical applications.

(3) The complete model performed well in all performance indexes in the ablation experiment, indicating its high generalizability in classification tasks. Removing the RNN encoding layer had the greatest impact on the model's performance, while removing the CNN and LSTM encoding layers had a relatively small impact, indicating that the RNN encoding layer in the model in this study had significant contributions to the model's performance, while the CNN and LSTM encoding layers had relatively small contributions.

In summary, the model in this study had optimal performance in classification tasks, and the RNN encoding layer played an important role in the model. At the same time, according to task requirements and the importance of performance indexes, appropriate pooling strategies could be selected to further optimize the model. These conclusions provide references for model design and optimization in practical applications.

6 REFERENCES

[1] Mustafidah, H., Suwarsito, S., Pinandita, T. (2022). Natural language processing for mapping exam questions to the cognitive process dimension. International Journal of Emerging Technologies in Learning, 17(13): 4–16. https://doi.org/10.3991/ijet.v17i13.29095

[2] Sadanandam, M. (2021). HMM based language identification from speech utterances of popular Indic languages using spectral and prosodic features. Traitement du Signal, 38(2): 521–528. https://doi.org/10.18280/ts.380232

[3] Pholo, M.D., Hamam, Y., Khalaf, A. B., Du, C.L. (2022). Differentiating between COVID-19 and tuberculosis using machine learning and natural language processing. Revue d'Intelligence Artificielle, 36(2): 313–318. https://doi.org/10.18280/ria.360216

[4] Mah, P.M., Skalna, I., Pełech-Pilichowski, T., Muzam, J., Munyeshuri, E., Uwakmfon, P. O., Mudoh, P. (2022). Integration of Sensors and Predictive Analysis with Machine Learning as a Modern Tool for Economic Activities and a Major Step to Fight Against Climate Change. Journal of Green Economy and Low-Carbon Development, 1(1): 16–33. https://doi.org/10.56578/jgelcd010103

[5] Mah, P.M. (2022). Analysis of artificial intelligence and natural language processing significance as expert systems support for e-health using pre-train deep learning models. Acadlore Transactions on AI and Machine Learning, 1(2): 68–80. https://doi.org/10.56578/ataiml010201

[6] Nabiilah, G.Z., Suhartono, D. (2023). Personality classification based on textual data using Indonesian pre-trained language model and ensemble majority voting. Revue d'Intelligence Artificielle, 37(1): 73–81. https://doi.org/10.18280/ria.370110

[7] Dhaka, S., Mathur, J., Wagner, A., Agarwal, G.D., Garg, V. (2013). Evaluation of thermal environmental conditions and thermal perception at naturally ventilated hostels of undergraduate students in composite climate. Building and Environment, 66: 42–53. https://doi.org/10.1016/j.buildenv.2013.04.015

[8] Wan Hamzah, W.M.A.F., Ismail, I., Yusof, M.K., Mohd Saany, S.I., Yacob, A. (2021). Using learning analytics to explore responses from student conversations with chatbot for education. International Journal of Engineering Pedagogy, 11(6): 70–84. https://doi.org/10.3991/ijep.v11i6.23475

[9] Praus, P. (2020). Information entropy for evaluation of wastewater composition. Water, 12(4): 1095. https://doi.org/10.3390/w12041095

[10] Bouaine, C., Faouzia, B., Imane, S. (2023). Word embedding for high performance cross-language plagiarism detection techniques. International Journal of Interactive Mobile Technologies, 17(10): 69–91. https://doi.org/10.3991/ijim.v17i10.38891

[11] Neiger, V., Rosenkilde, J., Solomatov, G. (2020). Generic bivariate multi-point evaluation, interpolation and modular composition with precomputation. In Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation, Kalamata Greece, pp. 388–395. https://doi.org/10.1145/3373207.3404032

[12] Ajallouda, L., Fagroud, F.Z., Zellou, A., Benlahmar, E.H. (2022). A systematic literature review of keyphrases extraction approaches. International Journal of Interactive Mobile Technologies, 16(16): 31–58. https://doi.org/10.3991/ijim.v16i16.33081

[13] Mann, S., Arora, J., Bhatia, M., Sharma, R., Taragi, R. (2023). Twitter sentiment analysis using enhanced BERT. In Intelligent Systems and Applications: Select Proceedings of ICISA 2022, pp. 263–271. https://doi.org/10.1007/978-981-19-6581-4_21

[14] Alissa, S., Wald, M. (2023). Text simplification using transformer and BERT. Computers, Materials & Continua, 75(2): 3479–3495. https://doi.org/10.32604/cmc.2023.033647

[15] Mustafidah, H., Suwarsito, S., Pinandita, T. (2022). Natural language processing for mapping exam questions to the cognitive process dimension. International Journal of Emerging Technologies in Learning, 17(13): 4–16. https://doi.org/10.3991/ijet.v17i13.29095

[16] Zhang, J., Pu, J., Xue, J., Yang, M., Xu, X., Wang, X., Wang, F.Y. (2023). HiVeGPT: human-machine-augmented intelligent vehicles with generative pre-trained transformer. IEEE Transactions on Intelligent Vehicles. 8(3): 2027–2033. https://doi.org/10.1109/TIV.2023.3256982

[17] Jiang, K., Zhu, M., Bernard, G.R. (2022). Few-shot learning for identification of COVID-19 symptoms using generative pre-trained transformer language models. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Grenoble, France, pp. 307–316. https://doi.org/10.1007/978-3-031-23633-4_21

[18] Mahmoudi, O., Bouami, M.F., Badri, M. (2022). Arabic language modeling based on supervised machine learning. Revue d'Intelligence Artificielle, 36(3): 467–473. https://doi.org/10.18280/ria.360315

[19] Sait, A.R.W., Ishak, M.K. (2023). Deep learning with natural language processing enabled sentimental analysis on sarcasm classification. Computer Systems Science & Engineering, 44: 2553–2567. https://doi.org/10.32604/csse.2023.029603

[20] Melethadathil, N., Heringa, J., Nair, B., Diwakar, S. (2019). Mining Inter-Relationships in Online Scientific Articles and its Visualization: Natural Language Processing for Systems Biology Modeling. International Journal of Online and Biomedical Engineering, 15(02): 39–59. https://doi.org/10.3991/ijoe.v15i02.9432

[21] Marzouk, R., Alabdulkreem, E., Nour, M.K., Al Duhayyim, M., Othman, M., Zamani, A.S., Motwakel, A. (2023). Natural language processing with optimal deep learning-enabled intelligent image captioning system. Computers, Materials & Continua, 74(2): 4435–4451. https://doi.org/10.32604/cmc.2023.033091

[22] Makridis, G., Mavrepis, P., Kyriazis, D. (2023). A deep learning approach using natural language processing and time-series forecasting towards enhanced food safety. Machine Learning, 112(4): 1287–1313. https://doi.org/10.1007/s10994-022-06151-6

7 AUTHORS

Lin Wang, PhD, is an associate professor, was selected as the high-level talent in Hainan Province. He is the creator of Neutralization System, Tai Chi Yoga, Class I instructor of National Health Qigong, a psychological counsellor, and a senior Yoga Instructor. He has more than ten years of experience in teaching Tai Chi abroad and is engaged in theoretical practice and research development of Tai Chi yoga, meditation and decompression, Health Qigong, physical and mental growth. He has published several papers at home and abroad, including 7 monographs and 2 papers included in international conferences, presided over 1 provincial-level project, and won 2 national second prizes and 1 third prize for his papers. He studied for a bachelor degree of Physical Education and Training from Hebei Sport University in 1994–1998, a master degree in the School of Physical Education of Hebei Normal University in 1999–2001, and a PhD of theories and methods of physical education, sports training, health care and adaptive sports in the Russian National University of Physical Education, Sports and Tourism in 2003–2007. He taught at the Department of Traditional Martial Arts, Hebei Institute of Physical Education in 1998–1999, at the School of Physical Education, Hebei Normal University in 2001–2002, and at the school of Sports Theory and Training, Russian Sports University in 2007–2015. Also, he worked as General Manager and Head Coach of Russian Tai Chi Center "Тайцзи Центр" in 2003–2016. Since 2016, he has worked in Hainan Vocational University of Science and Technology (email: wanglintaichi@163.com).

Weifeng Deng is a doctorate candidate in DPU. She graduated from the English Department of Hebei Normal University with a master degree in English Language and Literature in 2002. From April 2003 to November 2007, she worked as a part-time Chinese teacher in various schools and training courses in Moscow. After returning to China in 2007, She worked as an interpreter of English and Russian. From September 2014 to July 2015, she was an English teacher and counsellor in Hebei Foreign Studies University. She has worked in Hainan Vocational University of Science and Technology since 2016. She has published 6 papers, including 3 in international conferences, and has guided her students to the first prizes and second prizes in many national and provincial English competitions (email: 18976257477@163.com).