An Artificial Intelligence Method for Automatic Assessment of Fuzzy Semantics in English Literature

Meiyan LI
Faculty of International Studies
Henan Normal University
Henan 453007, China

***Abstract*—This Online writing and evaluation are becoming increasingly popular, as is automatic literature assessment. The most popular way is to obtain a good evaluation of the essay and article is by the automatic scoring model. However, assessing fuzzy semantics contained in reports and papers takes much work. An automated essay and articles assessment model using the long-short-term memory (LSTM) neural network is developed and validated to obtain an appropriate assessment. The relevant theoretical basis of the recurrent neural network is introduced first, and the quadratic weighted kappa (QWK) elevation method is cited here to develop the model. The LSTM network is then awarded for developing the general automatic assessment model. The available model is modified to get better performance by adding a convolutional layer(s). Finally, a data set of 7000 essays is segmented based on the ratio of 6:2:2 to train, validate, and test the model. The results indicate that the LSTM network can effectively capture the general properties of the essay and articles. After adding the convolutional layer(s), the LSTM+convolutional layer(s) model can get better performance. The QWK values are higher than 0.6 and have an improvement of 0.097 to 0.134 compared with the LSTM network, which proves that the results of the LSTM network combined with the convolutional layer(s) model are overall satisfactory, and the modified model has practical values.**

Keywords*—*Automatic assessment; recurrent neural networks; long short-term memory (LSTM); quadratic weighted kappa (QWK); convolutional layer

I. Introduction

In the traditional literature evaluation scene, the teacher or organizer designs the tasks, the candidates answer, and the teacher or organizer finally assesses the quality of the candidates' products by scoring. The most tedious part of the whole process is the manual evaluation and marking. It brings a heavy workload to evaluators and is prone to correction errors and long evaluation cycles, negatively impacting assessing. As computer technology and artificial intelligence develop, the automatic identification technology of filling multiple-choice questions in answer sheets has been widely used. The main task of assessment, involving complex processes such as handwritten digit detection, recognition and understanding, still relies on manual correction, especially in literature review and evaluation with fuzzy semantics. The candidates' handwritten text and the printed text of the paper are distributed in the same picture. When assessing, it is necessary to make an accurate distinction. At the same time, it is essential to detect the type of question to which the text belongs so that different correction strategies can be used according to various tasks. In addition, the handwriting of other candidates is different, and the readers are either compact or sparse, which makes text detection and recognition face greater challenges [1,2]. With the enrichment of computing and data resources and the innovation of various network structures, deep learning has performed well in various tasks, and therefore, artificial intelligence has gradually been widely used in images and text detection [3, 4]. The key technologies required to realize intelligent marking technology, including handwritten character detection, recognition, etc., are also very popular research fields. Recently, many relatively advanced research results have been reported, which makes it possible to achieve comprehensive intelligent evaluation [5, 6].

The continuous development of artificial intelligence and computer performance significantly improved the accuracy and efficiency of automatic essay and article evaluating and scoring models. Also, the development of neural networks has promoted the achievements of natural language processing technology in some related fields. It has brought new research directions to the automatic essay and articles evaluating and scoring research [7, 8]. The most typical is Automated Student Assessment Prize (ASAP) essay-scoring competition organized by Kaggle in 2012, and the appearance of the ASAP made public datasets available for many research efforts. Based on this dataset, researchers in different countries have conducted related research on essays and articles evaluating and scoring tasks. For instance, Alikaniotis D et al. (2016) [9] built a model modified from LSTM by citing the ASAP data set. Then they used this model for t automatic evaluation and scored essays and articles. Similarly, using the ASAP data set, Kavehet al. developed a model on the basis of recurrent neural networks and tried to use this model to learn the consistency between an essay and articles and real scores [10]. The results show that this method improved the model's ability to understand and capture the essay's main information and reports. Farag Y et al. also developed an automated easy evaluating and scoring model using the ASAP data set and the com. Theetween, a series of baselines and model outcomes, demonstrates its ability on the Automated Essay Scoring (AES) task and the flagging adversarial input, strengthening the effectiveness of neural essay and articles evaluating and scoring models [11]. These significant explorations kicked off the prelude of neural networks for automatic essays and papers evaluating and scoring.

Meanwhile, the successful application of large-scale pre-trained language models has brought breakthroughs to many natural language processing tasks as well. For instance, Rodriguez et al. compared the BERT and XLNet model based on natural language processing (NLP) neural networks adopted to achieve a high-quality Kaggle AES dataset. The results show that BERT and XLNet can produce more accurate outcomes than manual results and save time and money in grading essays and articles [12]. Mayfield and Black used the pre-trained language model BERT to solve the problem of automatic writing, paperless evaluating and scoring and their experiment. They showed that BERT achieved good results in the automated English essay and articles evaluating and scoring task, indicating that BERT is theoretically and practically feasible to solve the automatic grading of English papers and reports [13]. From the system structure level, it can be seen that using software algorithms to extract the information feature values of standard files is the foundation for achieving accurate translation in non-semantic environments. The similarity between words and sentences is a factor that causes the system to decline in translation in different semantic environments. The relative degree of a sentence includes multiple aspects such as part of speech, syntax, and sentence structure. By calculating sentence similarity, the degree of differentiation of sentences can be found. The larger the similarity value, the more information about word form, syntax, and semantics between the two sentences is resolved.

Through the analysis of the current research about automated essays and article evaluating and scoring, the development of automated essay and article evaluating and scoring technology can be divided into three periods [14,15]. The first period is the artificial feature extraction period. The typical feature of this period is determining which parts of the essay and articles to extract manually and building a regression model based on these features by machine learning-based methods [16]. The advantage of this method is that it can be logically explained. The disadvantage of this method is also obvious. Due to many relevant features that need to be extracted by humans, it is challenging to reconstruct a large number of data sets. At the same time, these features may not directly reflect the deep fuzzy semantics of the essay and articles. The second period is the neural network period. In this period, word vectors are used to build models. Word vectors are developed by learning text information from essays. The main advantage is that it only needs manual feature extraction work. The disadvantage is that the model is not open and transparent since the model's training process is carried out in a black box. The third period is the transfer learning period [17, 18]. In this period, the knowledge learned from a large amount of corpus using the pre-trained language model is transferred to the automatic evaluating and scoring task, which can effectively avoid training the model from the beginning and achieves good results with relatively small datasets.

The application of distributed ultra-large computing power computers and the advent of large-scale data sets jointly promote the further development of neural networks in the field of automatic evaluating and scoring of essays and articles and make the automated essay and articles and articles evaluating and scoring evolve into the third stage as described above. Although the LSTM network is more efficient than the standard recurrent neural network and can effectively improve the model learning ability, capture the features and provide generally acceptable results to some extent. Still, it may not work well when meeting with fuzzy semantics contained in literature. Therefore, in this work, the main model modified from LSTM networks is developed and improved by adding convolutional neural network layers since the introduction of a convolutional neural network layer into the evaluating and scoring model can strengthen the ability of the model to capture local information in the essay and articles. Then the effectiveness of the LSTM networks combined with convolutional neural network layers is evaluated with a relatively large database.

In the process of English translation, the more similar the semantics are, the greater the correlation, which can easily lead to comprehension errors in different contexts and bring difficulties to translation work. Based on this, a similarity calculation model based on a combination of semantic dictionaries and corpora is established, starting from bilingual materials of English and Chinese sentences. Under the conditions of the established corpus, relevant semantic extraction rules and dependencies are determined. Through the English sentence similarity algorithm and the vector space model standard, the calculated similarity is used as a vector element to find the degree of differentiation in sentences, distinguishing between sentences and words in terms of part of speech, syntax, and tense. The research results indicate that the system has high accuracy and recall rate in sentence translation, especially in the English translation process of prepositions, function words, and tenses, with higher translation efficiency and accuracy. Evaluating the fuzzy semantics contained in reports and papers requires a lot of work. This article develops and validates an automated paper evaluation model using Long Short Term Memory (LSTM) neural networks to obtain appropriate evaluations.

This article evaluates the innovation contribution as follows:

1) Developed and validated an automated paper evaluation model using Long Short Term Memory (LSTM) neural networks to obtain appropriate evaluations. By adding convolutional layers, modifications were made to the available models to achieve better performance.

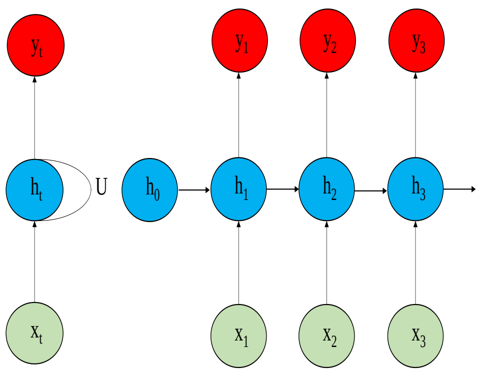
2) Traditional RNNs are prone to gradient vanishing when dealing with long sequences, making it difficult to train. LSTM introduces a gating mechanism, which can effectively alleviate the problem of gradient vanishing and process longer sequence data.

3) LSTM can better capture long-term dependencies in sequence data through cell state and gating mechanisms.

II. Theoretical basis of Recurrent Neural Network

A. Recurrent Neural Network

Recurrent Neural Networks (RNNs) are mainly used to deal with timing problems. There is a connection between different neuron states in the RNN hidden layer. The input in the neuron state includes the input layer and consists of the outcomes of the previous neuron state. The semantic information before and after the text sequence is very important to the recognition accuracy of the current series, so RNN is mostly used in text recognition, speech recognition and other tasks. RNNs are also used in artificial intelligence and machine learning. In the development of the automatic essay and articles evaluating and scoring model, essay and articles text is regarded as sequence data because the text information contains contextual semantic information. RNNs have advantages in processing data, and RNNs have advantages in processing textual information because RNNs can memorize previously learned content [19].



(a) General structure of RNNs (b) Expended structures of RNNs

Fig. 1. The model structure of the recurrent neural network.

Text information is generally input with unequal lengths, but the word length of each text is different. At the same time, understanding the text information needs to consider the order of words; that is, words are input individually. The recurrent neural network can also accept sequence data with an unequal length not defined in advance [20]. The performance of RNNs is affected by the knowledge learned in the past, which means that the processing output may obtain the features of previously known knowledge. Remembering the features learned from the previous knowledge through the vector representation of the hidden layer, the RNNs model can generate one or more output vectors from one or more input vectors and generate different outputs even with the same input. The recurrent neural network model's structural characteristics are shown in Fig. 1(a).

The figure above depicts the properties of the recurrent neural network at sequence number t. W, U, and V correspond to the weights of the input layer x_t passed to the hidden layer h_t, between the hidden layers, and from the hidden layer h_t to the output layer y_t, respectively. As can be seen from Fig. 1(b), an important feature of RNNs is parameter sharing, and the meanings of parameters are:

1) x_t is the input of the training sample, and t is the sequence number;

2) h_t is the hidden state of the model, and it is computed by x_t and h_{t-1};

3) y_t is the model's output, which is determined by the h_t at the current hidden state.

The RNNs equation is expressed as follows:

$$y_t = g(Vh_t) \tag{1}$$

$$h_t = f(Wx_t + Uh_{t-1}) \tag{2}$$

RNNs can memorize knowledge, but they need help remembering long text information. This is because when ordinary RNNs deal with long sequence problems, passing the gradient from the back of the sequence to the front line through the back-propagation algorithm is difficult. In recent decades, many versions have been developed and built with the research progress of the recurrent neural network, among which the LSTM network is the most successful. This is because the LSTM network solves the long-term dependence problem in the training process.

B. LSTM Network

During the predicting process, the recognized text could be infinite because of the variety of the length of the text sequence. Suppose the position of the currently predicted text or the text whose semantic information is consistent with the text is large. In that case, the ordinary recurrent neural network will lose part of its semantic information, thereby reducing the accuracy of prediction results. The network of LSTM is a more advanced version of RNNs and solves this problem very well. The setting of the gate structure prevents the LSTM model from retaining all the information indiscriminately, like the RNN model. It cannot highlight the key points which solve the problem of gradient disappearance in the learning process of the model [21]. The main difference between LSTM and ordinary recurrent neural networks is that LSTM has three gates: the forgotten, input, and output. The model can remind the information of each previous neuron, reducing the loss of information in the transmission process. Therefore, the LSTM model can efficiently retain important data according to the task objective and uses the information judgment as to the parameter of model learning, which greatly increases the learning efficiency of the model [22].

At the same time, LSTM can also solve the problem of gradient descent and thus is widely used in sequence data tasks, such as language understanding, segmentation, and translation. It has been verified in many experiments that LSTM is more effective than standard RNNs, which can effectively improve the model learning ability. Due to the careful design of LSTM, introducing the gating mechanism can alleviate the gradient disappearance of the recurrent neural network, thereby memorizing long text information. The LSTM consists of five different parts [23, 24]:

1) Unit state: The internal memory of the LSTM unit.

2) Hidden state: This state is used to calculate the prediction result of the model.

3) Input gate: It is used to judge the amount of inputted information that can be sent to the unit state.

4) Forget gate: It is used to judge the number of previous unit states that can be sent to the current unit state.

5) Output gate: It judges how many unit states can be sent to the hidden state.

In a recurrent neural network model, the unit's state changes with each input in the current form. This directly leads to the fact that in the cyclic neural network structure, the team's state is always changing, and this mechanism causes the cyclic neural network to not store long-term dependencies well. In LSTM, the current cell state relies not only on the input of the current state but also that of at previous step. Therefore, the LSTM can decide to update at a certain moment or forget the internal information stored in each neuron in the cell state. Namely, the LSTM has a mechanism to keep the cell state unchanged so that the LSTM can store long-term dependencies. The data flow diagram in LSTM is shown in Fig. 2.

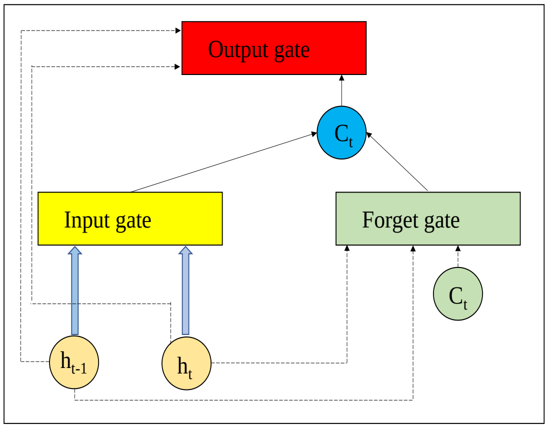


Fig. 2. The data flow diagram in LSTM.

As shown in Fig. 2, LSTM mainly introduces the gating unit concept to control the unit's state. LSTMs have gates for each operation a cell needs to perform. Each gating team takes a continuous value ranging from 0 to 1, where 0 indicates that all the information is blocked and will not be passed to the next step, and 1 means that all information is passed to the next gate. LSTM network uses such gating units for each neuron in the team. The calculation equation of each gate control unit is described below. The calculation equation for the input gate is shown in the following equations.

$$i_t = a(w[h_{t-1}, x_t]^T + b_i) \quad (3)$$

$$\tilde{c}_t = tanh(\psi_{cyt}) - tanh(w_c[h_{t-1}]^T + b_c) \quad (4)$$

where, i_t = represents the information amount of the input state saved to the unit state, and $\tilde{c}_t$ means the information after the Unit State is updated. *A* is defined as an activation function, the weight matrices are ω, and ω_c and the bias matrices are b_i and b_c. The revised equation for the current cell state is shown in Eq. (5).

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \quad (5)$$

In the equation, c_t represents the information currently saved by the unit state, and c_{t-1} is the information held by the unit state at the previous moment. The calculation equation of the forgetting gate is shown in Eq. (6).

$$f_t = a\left(w_f[h_{t-1}, x_t]^T + b_f\right) \quad (6)$$

where, f_t indicates the forget gate at time t. The forget gate controls whether the information can be transmitted to the current moment. Calculate f_t is calculated in the matrix formed by the output state of the hidden layer h_{t-1} and the current input state x_t at the previous input, multiply it with the weight matrix ω_f, and finally, add the bias matrix b_f. The calculation equation of the output gate is shown in Eq. (7) and Eq. (8).

$$f_t = a(w_0[h_{t-1}, x_t]^T + b_0) \quad (7)$$

$$h_t = u_t \, tanh(c_t) \quad (8)$$

In the equations, u_t is the updated output information of the storage unit, h_t is the output information of the hidden layer at the current moment, w_o is the weight used to calculate the forgotten team, and b_o is the bias matrix.

III. Model Construction and Experimental Setup

Based above research, the automatic evaluating and scoring model established is developed. The main model structure of the proposed model can be briefly described by following steps: first, the text in the corpus is segmented, and then the word segmentation result is used to construct a word vector, and the word vector work as the input of the LSTM evaluating and scoring model. At the same time, a convolutional layer is added before the LSTM layer. This can improve the prediction of the LSTM method because the convolutional layer can effectively understand the text's local information and fuzzy semantics.

A. LSTM-based Essay and Articles Automatic Evaluating and Scoring Model

Since the essay and article text contains contextual knowledge, it can be regarded as sequence information. The model-building process is shown in Fig. 3. Because the long short-term memory network cannot directly take text as input, the next step is to convert the essay and article text into word vectors after the intake of the corpus. Then the obtained word vectors will be passed to the neural network model for prediction, optimize the model through the optimizer, and finally build, the model.

The Adam optimizer is used in this experiment. In the model optimization, the mean squared error (MSE) loss function is selected to make the neural network model better optimized, which is shown in Eq. (9).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_t)^2 \quad (9)$$

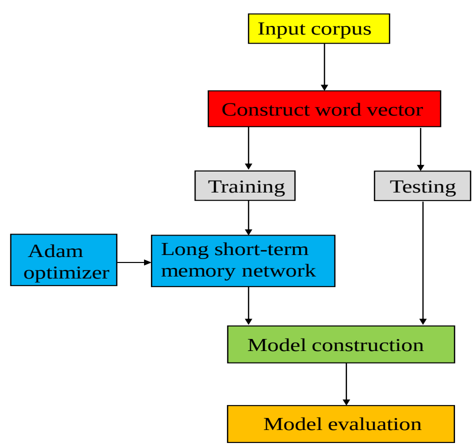


Fig. 3. The model-building process.

B. Evaluation Method

The QWK coefficient is generally used as an evaluation index in the automatic grading of related essays and articles. It can well reflect the consistency between the predicted quality index (mostly in the form of score) and the actual quality index of the essay and articles [25] and improves the validity of the evaluation results. The QWK coefficient introduces a penalty mechanism based on the KAPPA coefficient. If the error between the composition's predicted quality index and the composition's actual quality index is larger, the QWK coefficient will be smaller. The penalty mechanism enables the QWK coefficient to reflect the consistency between the quality index predicted by the proposed model and the manual quality index of the essay and articles. Although some algorithms have good performance in certain datasets, they may encounter difficulties in other datasets. In addition, by adjusting the hyperparameters of the algorithm that controls the training process, performance can be improved. The performance of models in machine learning problems largely depends on the dataset and training algorithms. Choosing the correct training algorithm can change the story of the model. Nowadays, an adaptive machine learning algorithm is used for different biological, biomedical, and natural categories [26].

The fuzzy semantic of English literature is a two-sided existence, such as some ambiguous expressions. And due to the differences between cultures, clearer semantic content can result in differences in understanding. The QWK coefficient uses a weighting method to strengthen the result. If the excessive deviation between the model prediction and real quality index, it would penalise the extreme difference by reducing the QWK coefficient. Using the QWK coefficient can not only measure the consistency of the evaluating and scoring model but also improve the recognition accuracy of fuzzy semantic contents in English literature. The equation for calculating the QWK coefficient is shown in Eq. (11). But the first step is to construct the weight matrix, and the calculation equation of the weight matrix is shown in Eq. (10).

$$W_{i,j} = \frac{(i-j)^2}{(n-1)^2} \quad (10)$$

In the equation, i and j is the quality index given by the experts (real quality index) and essay and articles evaluating and scoring models, respectively. N indicates 9, the number of essays and articles database. After building the weight matrix, the matrix N and the prediction matrix E are built. $N_{i,j}$ is the amount samples with real rating i and model rating j. E is the outer product of the real marks and the model quality index vector. The QWK coefficients are calculated using the constructed E matrix and N matrix.

$$k = 1 - \frac{\sum_{i,j}(W_{i,j}N_{i,j})}{\sum_{i,j}(W_{i,j}E_{i,j})} \quad (11)$$

Evaluators give the essay and article quality index used in this paper, and the essay and articles' real quality index are regarded as a true index of essay and article quality. So, the evaluation of the evaluating and scoring model built in this paper is transformed into the consistency problem between the model results and the real marks. In this paper, the QWK coefficient is used as the measurement method to test the consistency since the QWK coefficient can reflect the model prediction quality index, which can reflect the capability of assessing the English literature with fuzzy semantics. The meaning of the QWK coefficient is shown in Table I.

TABLE I. THE MEANING OF QWK

QWK	Consistency strength
<0.20	Poor
0.21~0.40	General
0.41~0.60	Medium
0.61~0.80	Good
0.81~1.00	Best

This article collects a set of paper data, including papers from various disciplines. Firstly, we use a portion of the dataset for training, and then test the remaining dataset. Table II shows the performance of our model on the test set.

TABLE II. PERFORMANCE INDICATORS OF THE MODEL ON THE TEST SET

Accuracy	0.85
Precision	0.83
Recall	0.82
F1 Score	0.80

IV. EXPERIMENT AND RESULT ANALYSIS

A. Test Setup

1) Normalization and inverse normalization: The output of the proposed model is the predicted essay and article quality index. If the input essay and article quality index are in a different grade range, it may impact the model's outcomes.

In most situations, the quality index of essays and articles by scoring can be divided into three categories: the first is 40-95, the second is 65-95, and the third is 50- 95 points. Due to the different range of scores, the normalization of essay and articles scores is firstly performed to get a uniform index, and the core idea of normalization is to map essay and articles scores from different score segments to floating-point numbers between (0, 1), which is more conducive to processing of the model. The normalized calculation equation is shown in Eq. (12).

$$x' = \frac{x - x_{Min}}{x_{Max} - x_{min}} \quad (12)$$

In Eq. (12), x represents the real essay and articles score, M_{in} and M_{ax} represent the lowest and highest score of the scoring range, and x' represents the normalized result obtained by normalizing the real essay and articles score.

Before inputting into the model, the essay and article scores in the corpus are first normalized. Then the corpus is divided into a training set, validation set, and test set according to the ratio of 6:2:2. In the model's training process; the normalized score will be used as the label. After the training process is completed, the essay and articles in the test set are used as the input, and the trained model is then used to predict the score of the essay and articles. However, the score obtained at this state is not (0-100) but the normalized predicted score. In order to calculate the QWK coefficient between the real score results and the model prediction results, we need to perform inverse normalization on the normalized prediction scores. The inverse normalization calculation equation is shown in Eq. (13).

$$y = y'(xmin_{max} + x_{min}) \quad (13)$$

In Eq. (13), y represents the normalization result predicted by the model, and x_{min} and x_{max} represent the lowest and highest scores in the real scoring range. After performing the inverse normalization operation, the prediction score of the model can be obtained.

2) Experimental arrangement and hyperparameter settings: During the experiment, 7000 essays and articles corpus from the Kaggle dataset were cited and divided into the training set, validation set, and test set with the ratio of 6:2:2. To train the LSTM model on the training set, different numbers of hidden layers were used to observe the effect of other numbers of hidden layers on the results of the model b_v. The LSTM network structure used is shown in Fig. 4.

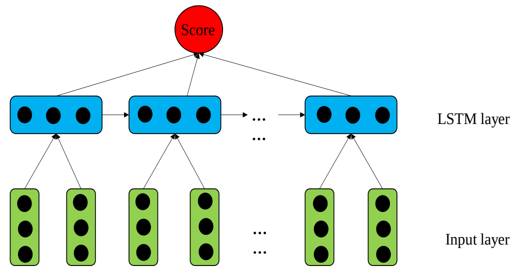


Fig. 4. The LSTM network structure.

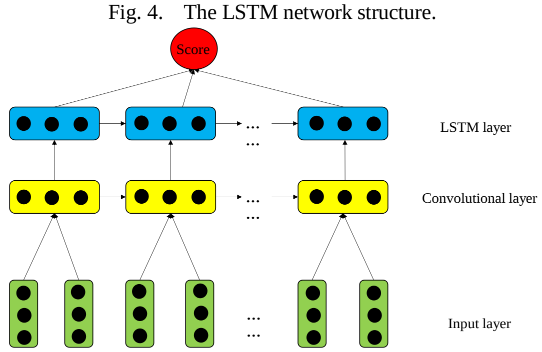


Fig. 5. The LSTM+ convolutional layer.

At the same time, in the research of Taghipour K et al [10]., it was found that adding a convolutional layer before the recurrent neural network can effectively improve the model's capability of capturing the local information of the text. Therefore, based on the above experiments, adding a convolutional layer before the LSTM layer is embedded and observing whether the additional convolutional layer(s) can improve the model's performance is proposed. The network structure of the LSTM+convolutional layer used in this study is shown in Fig. 5.

Simulation experiments determine the model's hyperparameters after building the LSTM model. The parameters are shown in Table III.

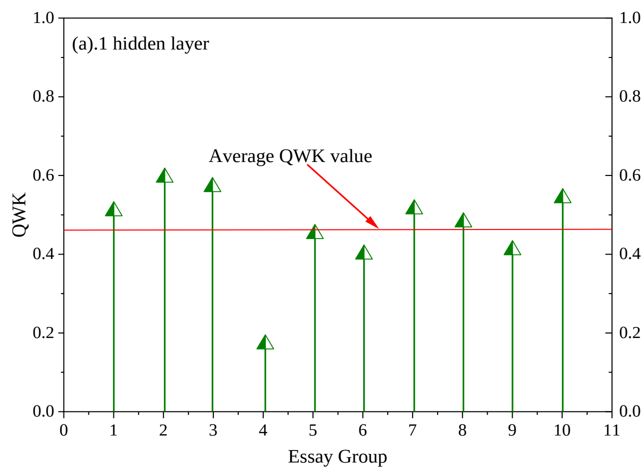
TABLE III. THE VALUES OF MODEL PARAMETERS

Parameters	Values
Learning rate	0.001
Batch size	64
Regular optimization	0.5
Activation function	ReLU
Number of iterations	30

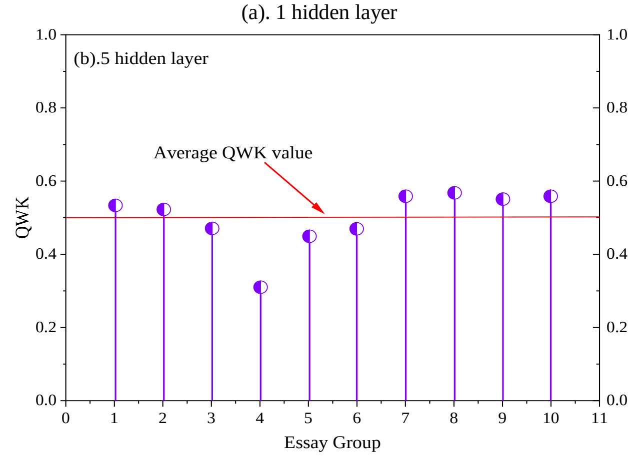
B. Results and Analysis

Based on completing the model construction in the previous section, the number of hidden layers of the network to 1, 5, and 9 layers are built. The QWK results obtained by the LSTM network model are shown in Fig. 6.

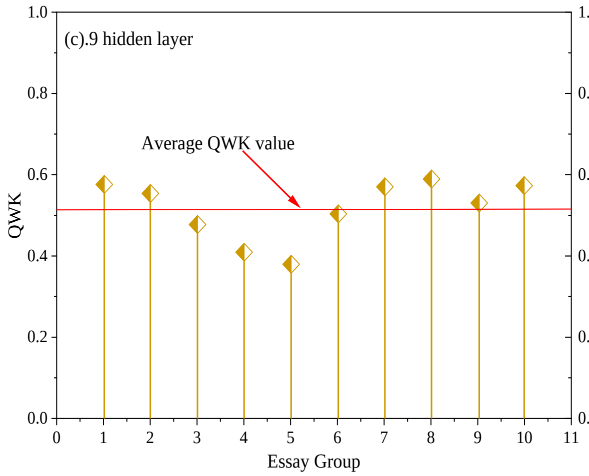
Fig. 6 shows that the QWK coefficient of the experimental results obtained by the essay and articles evaluating and scoring model constructed based on the LSTM network is at a moderate level of consistency, according to Table I. The value of QWK ranges from 0.17 to 0.59, which indicates that the accuracy of the LSTM is not satisfactory, as shown in Fig. 6(a). Increasing the number of hidden layers, the model's performance also increases. When the hidden layers are 5 and 9, the value of QWK are 0.31~0.57 and 0.38~0.59, respectively. The average weight of QWK also increases from 0.464 to 0.499 and 0.515 when hidden layers are increased from 1 to 5 and 9. It indicates that as the depth of the LSTM network grows, the model has better nonlinear expression ability and can perform more complex transformations. The experimental results show that deepening the network depth can effectively improve the output results.



(a). 1 hidden layer



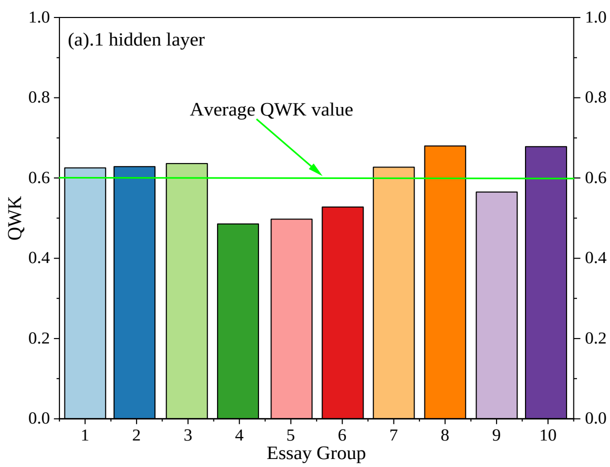
(b). 5 hidden layers



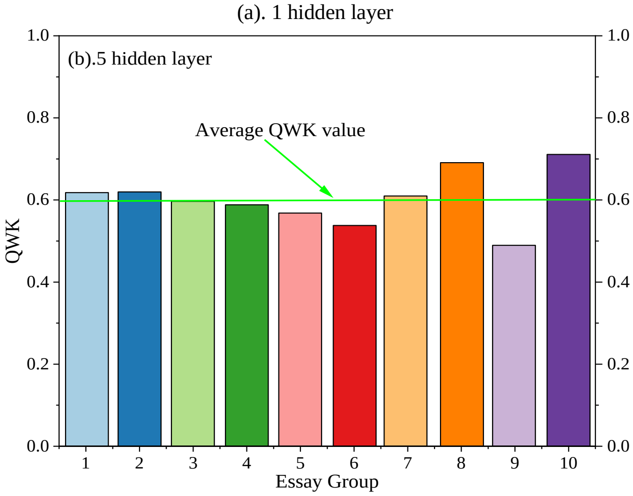
(c). 9 hidden layers

Fig. 6. The QWK values obtained by the LSTM network model.

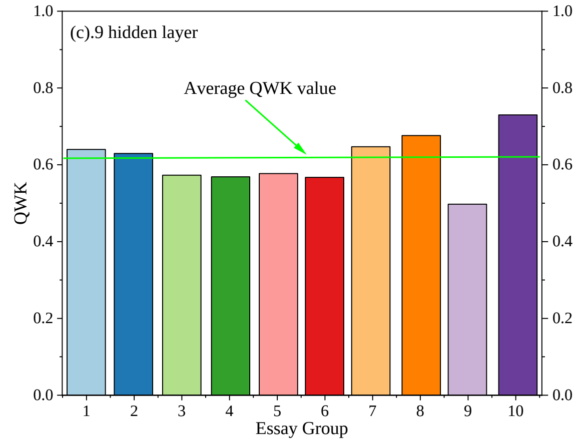
The convolutional layer is added to conduct a comparative neural network experiment based on the above experimental results. The addition of the convolutional layer is to improve the ability to capture the local text information. The QWK results obtained by adding the convolutional neural layer into the Long-Short-Term Memory Network model are shown in Fig. 7.



(a). 1 hidden layer



(b). 5 hidden layers



(c). 9 hidden layers

Fig. 7. The QWK results after adding the convolutional neural layer.

It can be seen from Fig. 7 that using the essay and articles evaluating and scoring the model constructed based on the LSTM network, adding the cumulative neural network layer has significantly improved the experimental results of the model. However, the QWK coefficient could be at a better level of consistency. After LSTM is incorporated with the convolutional neural layer, the value of QWK in 1, 5 and 9 layers are 0.41~0.68, 0.49~0.69 and 0.5~0.68, respectively. The average values of QWK are at a good level of consistency with values of 0.598, 0.604 and 0.612. This experimental result confirms that adding the convolutional layer can capture the local information of the text.

From Fig. 6 and Fig. 7, it is clear that the Long-Short-Term Memory Network model can provide an acceptable experimental result using the QWK coefficient in the evaluation process. Compared Fig. 6 with Fig. 7, adding the convolutional layer(s) can increase the accuracy of the proposed model. By comparing the average QWK values of LSTM and LSTM+ convolutional layer, it can be observed that the QWK values are improved (the improvement is 0.097~0.134), as shown in Fig. 8, indicating that the neural network with the convolutional layer(s) can perform more complex feature learning, and the network has better representation ability. Based on TableI, the LSTM+C can obtain good results and has practical values.

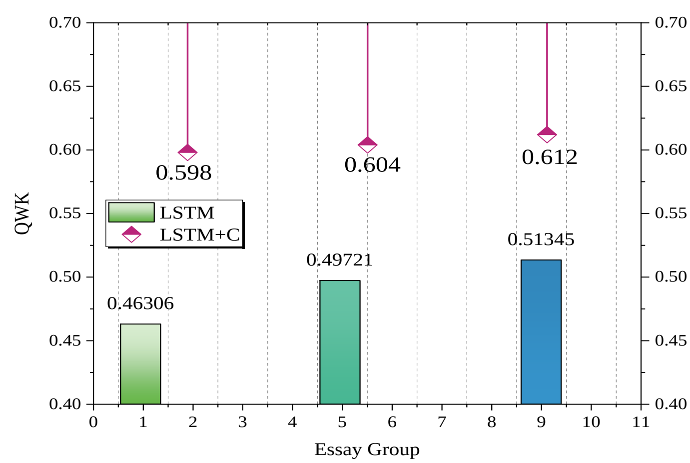


Fig. 8. The comparison of LSTM and LSTM+ convolutional neural layer.

The research results indicate that using papers and articles to evaluate and grade models constructed based on LSTM networks. This not only increases the cumulative neural network layer, but also significantly improves the experimental results of the model. The calculation for LSTM is feasible. Approximate calculation methods and other dependency methods can be used to accelerate the training process. For example, techniques such as truncation or compression can be used to reduce the number of parameters and calculate Type 2 simplification: in order to solve the problem of STM interpretation. A simpler model can be used, which has fewer parameters and mechanisms than LSTM, but still handles sequence numbers well. If the number of training is insufficient, it is possible to use the number of strong techniques to generate more books, or use transfer learning to utilize data from other tasks to improve model performance.

V. CONCLUSION

In this paper, the model construction process and related experimental settings are introduced， and the automatic essay and articles assessment model modified from the neural network is developed and validated in assessing English literature with fuzzy semantics. After comparing the results of the prediction, the following conclusions are drawn:

First, in developing the the model, the QWK elevation method is cited due to its consistency evaluation properties that that can improve the recognition of fuzzy semantics content. And then, the long-short-term memory network is modified by adding the convolutional layer(s) into the LSTM model.

Then, 7000 articles were segmented based on the ratio of 6:2:2 to train, validate, and test the proposed model. The comparison and analysis of experimental results indicate that LSTM can generally capture the real features of the articles. After adding the convolutional layer(s), the proposed model can get better performance by improving 0.097 to 0.134 of the QWK value. These results prove that the automatic essay and articles evaluating and scoring model based on neural network+ convolutional layer(s) can evaluate the English literatures containing fuzzy semantics.

However, an important advantage of neural networks is that they can learn complex patterns from a large amount of data. However, one limitation of this method is that it usually only captures the surface features of the data and may not be able to deeply understand the true meaning of the text. For English literature, this issue may be more severe due to the inherent ambiguity and ambiguity in language. Although English has relatively fixed grammar and vocabulary, the same word or phrase may have completely different meanings in different contexts, and this meaning usually requires in-depth contextual understanding to accurately grasp. Future research can focus on developing more effective models, such as combining natural language processing (NLP) technology and deep learning technology, to improve the model's ability to understand semantics while reducing reliance on a large amount of high-quality data.

FUNDING

This work was supported by the Education Science Research Foundation of Henan Normal University, Research on the Optimization of English Teaching Ecology in High Schools of Henan Province from the Perspective of Discipline Core Literacy (No.: 2021JK09).

REFERENCES

[1] Yannakoudakis H, Andersen ØE, Geranpayeh A, Briscoe T, Nicholls D. Developing an automated writing placement system for ESL learners. Applied Measurement in Education. 2018;31:251–67.

[2] Beseiso M, Alzubi OA, Rashaideh H. A novel automated essay scoring approach for reliable higher educational assessments. J Comput High Educ. 2021;33:727–46.

[3] Moreno A, Redondo T. Text analytics: the convergence of big data and artificial intelligence. IJIMAI. 2016;3:57–64.

[4] Lee Y-J, Park J-Y. Identification of future signal based on the quantitative and qualitative text mining: a case study on ethical issues in artificial intelligence. Qual Quant. 2018;52:653–67.

[5] Ramesh D, Sanampudi SK. An automated essay scoring systems: a ystematic literature review. Artif Intell Rev. 2022;55:2495–527.

[6] 6. Yang L, Liu W. Design of English intelligent simulated paper marking system. Complexity. 2021;2021:1–10.

[7] Dong F, Zhang Y, Yang J. Attention-based recurrent convolutional neural network for automatic essay scoring. Proceedings of the 21st conference on computational natural language learning (CoNLL 2017). 2017. p. 153–62.

[8] Uto M, Okano M. Robust neural automated essay scoring using item response theory. Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part I 21. Springer; 2020. p. 549–61.

[9] Alikaniotis D, Yannakoudakis H, Rei M. Automatic text scoring using neural networks. arXiv preprint arXiv:160604289. 2016;

[10] Taghipour K, Ng HT. A neural approach to automated essay scoring. Proceedings of the 2016 conference on empirical methods in natural language processing. 2016. p. 1882–91.

[11] Farag Y, Yannakoudakis H, Briscoe T. Neural automated essay scoring and coherence modeling for adversarially crafted input. arXiv preprint arXiv:180406898. 2018;

[12] Rodriguez PU, Jafari A, Ormerod CM. Language models and automated essay scoring. arXiv preprint arXiv:190909482. 2019;

[13] Mayfield E, Black AW. Should you fine-tune BERT for automated essay scoring? Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications. 2020. p. 151–62.

[14] Ke Z, Ng V. Automated Essay Scoring: A Survey of the State of the Art. IJCAI. 2019. p. 6300–8.

[15] Hussein MA, Hassan H, Nassef M. Automated language essay scoring systems: A literature review. PeerJ Comput Sci. 2019;5:e208.

[16] Uto M. A review of deep-neural automated essay scoring models. Behaviormetrika. 2021;48:459–84.

[17] Lagakis P, Demetriadis S. Automated essay scoring: A review of the field. 2021 International Conference on Computer, Information and Telecommunication Systems (CITS). IEEE; 2021. p. 1–6.

[18] Madnani N, Cahill A. Automated scoring: Beyond natural language processing. Proceedings of the 27th International Conference on Computational Linguistics. 2018. p. 1099–109.

[19] Cai C. Automatic essay scoring with recurrent neural network. Proceedings of the 3rd International Conference on High Performance Compilation, Computing and Communications. 2019. p. 1–7.

[20] Nadeem F, Nguyen H, Liu Y, Ostendorf M. Automated essay scoring with discourse-aware neural models. Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications. 2019. p. 484–93.

[21] Zazo R, Lozano-Diez A, Gonzalez-Dominguez J, T. Toledano D, Gonzalez-Rodriguez J. Language identification in short utterances using

long short-term memory (LSTM) recurrent neural networks. PLoS One. 2016;11:e0146917.

[22] Cai W, Cai Z, Zhang X, Wang X, Li M. A novel learnable dictionary encoding layer for end-to-end language identification. 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2018. p. 5189–93.

[23] Liang G, On B-W, Jeong D, Kim H-C, Choi GS. Automated essay scoring: A siamese bidirectional LSTM neural network architecture. Symmetry (Basel). 2018;10:682.

[24] Xia L, Guan M, Liu J, Cao X, Luo D. Attention-based bidirectional long short-term memory neural network for short answer scoring. Machine Learning and Intelligent Communications: 5th International Conference, MLICOM 2020, Shenzhen, China, September 26-27, 2020, Proceedings 5. Springer; 2021. p. 104–12.

[25] Uysal I, DOĞAN N. How Reliable Is It to Automatically Score Open-Ended Items? An Application in the Turkish Language. Journal of Measurement and Evaluation in Education and Psychology. 2021;12:28–53.

[26] Nematzadeh, S. , Kiani, F. , Torkamanian-Afshar, M. , & Aydin, N. . (2022). Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: a bioinformatics study on biomedical and biological cases. Computational biology and chemistry(97-), 97.