IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 5 January 2023, accepted 16 March 2023, date of publication 14 April 2023, date of current version 19 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3260840

RESEARCH ARTICLE

A Cross-Lingual Hybrid Neural Network With Interaction Enhancement for Grading Short-Answer Texts

YISHAN CHEN[1], JIANHUA LUO[1], XINHUA ZHU[2], HAN WU[2], AND SHANGBO YUAN[3]
[1]School of Business, Guilin Tourism University, Guilin 541006, China
[2]School of Computer Science and Engineering, Guangxi Normal University, Guilin 541004, China
[3]TUM School of Engineering and Design, Technical University of Munich, 85748 Munich, Germany

Corresponding author: Jianhua Luo (43877557@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Contract 62062012 and Contract 61967003, and in part by the Natural Science Foundation of Guangxi of China under Contract 2020GXNSFAA159082.

ABSTRACT Automatic Short-Answer Grading (ASAG) is an application for recognizing textual entailment in smart education. With the continuous expansion of the application scope of artificial neural networks, many deep learning models have been applied to grading short-answer texts. However, the coding structures and interaction forms of existing models are still too simple to meet the requirements of the ASAG task, resulting in low scoring accuracy. To address these challenges, we propose a cross-lingual hybrid neural network with interaction enhancement for ASAG. First, we sequentially use a convolutional neural network and bidirectional Long Short-Term Memory (LSTM) network to encode the answer text. Then, we introduce an interaction enhancement layer consisting of reference-answer-to-student-answer and student-answer-to-reference-answer attentions, and we combine the attentions and their inputs to form enhanced representations of answer texts. Finally, we introduce two Siamese Bi-LSTM networks to fuse the enhanced representations of answer texts and combine their multiple pooled vectors for grade classification on a multi-linear prediction layer. The experimental results show that our model significantly improves the performance of various simple models for Chinese and English ASAG tasks. The code is available online at https://github.com/wuhan-1222/DL_ASAG.

INDEX TERMS Automatic short-answer grading, textual entailment, hybrid neural networks, interaction enhancement.

I. INTRODUCTION

Automatic Short Answer Grading (ASAG) is a key component of Intelligent Tutoring Systems (ITSs) [1], which can capture the current cognitive level of students and provide important clues for the system to formulate personalized learning.Because both students' and reference answers are natural language texts, ASAG is regarded as an application for Recognizing Textual Entailment (RTE) in smart education [2], as shown in Table 1. Feature engineering was the dominant technique in most early ASAG methods [2], [3], [4], [5], [6], [7], [8], [9], [10]. Since then, many traditional machine learning methods using manual features have emerged. Examples include using token overlap to represent text similarity [2], [6], [7], [10]; using syntax and

TABLE 1. An example of ASAG in the english mohler dataset [3].

QUESTION	What is a variable?
REF. ANS.	A location in memory that can store a value.
Score value	0-5
STUD. ANS.	(1) Variable can be a integer or a string in a program. **(score=2)** (2) A variable is a location in memory where a value can be stored. **(score=5)** (3) A named object that can hold a numerical or letter value. **(score=3.5)**

The associate editor coordinating the review of this manuscript and approving it for publication was Junhua Li.

dependency to analyze and represent the answer text [3], [5]; using tf-idf [4], [7], [8], [11] to form the answer text vector; and utilizing the WordNet knowledge base [6], [7], edits [9] or sentence embedding [2], [5], [7] to computing text similarity. However, feature-based machine learning methods have many shortcomings. First, some manual features, such as WordNet knowledge, are very large and sparse, and there is a huge labor cost to build them. Second, feature extraction requires some pre-treatment steps, such as lemmatization, token boundary, part-of-speech tagging, and dependency parses, while each pre-treatment step may cause certain errors that result in error transmission and accumulation. Moreover, feature engineering lacks effective methods to represent text timing sequences and cannot effectively encode the context of answer texts.

With the continuous expansion of the application scope of artificial neural networks, many deep learning models, such as LSTM-based [12], [13], [14], [15], Convolutional Neural Network (CNN) & LSTM-based [16], and Transformer-based models [17] have been applied to grading short answer texts. These deep learning models are divided into two categories: one type has no standard answers [12], [14], [16], such as composition or essay, so a separate corpus and training for each question is required; the other has standard answers, such as short-answer questions [13], [15], [17], which provides corpora and training for all questions uniformly. This study focuses only on the latter type of task with a standard answer. Considering that the length and sequence of the text of students' answers may be different, the deep learning method for ASAG is required to combine multiple neural networks to encode the answer text to implement the deep interaction between the student's answer and the standard answer. Unfortunately, the coding structures [11], [12], [13], [14], [16] and interaction forms [11], [12], [13], [14], [15] of the existing models are still too simple to meet the requirements of the ASAG task, resulting in low scoring accuracy. Therefore, the construction of an advanced deep learning model with a more complex coding structure and interaction form is a major challenge for current ASAG research.

In this paper, to address the above challenge, we take advantage of hybrid neural networks for natural language understanding [18] and propose a cross-lingual deep learning model using hybrid neural networks for grading short-answer texts. The main contributions of this study are summarized as follows:

(1) Based on the characteristics of the ASAG task, we propose a hybrid neural network model consisting of a CNN phrase layer, Bi-LMST encoding layer, attention-based interaction layer, Bi-LMST fusion layer, and multi-linear prediction layer. The model significantly improves the existing simply structured models [12], [13], [14], [15], [17], can span different language environments, and achieves excellent performance in both Chinese and English ASAG tasks.

(2) We use two Siamese CNNs to extract local features from the input sequences to form phrases in the answer texts. This is especially important for Chinese answers as it is equivalent to the word segmentation process of Chinese text.

(3) We introduce an interaction enhancement layer consisting of reference-answer-to-student-answer and student-answer-to-reference-answer attentions, and we combine these attentions and their inputs through various forms, such as concatenation, difference, and product, to form the enhanced representations of answer texts.

(4) We introduce two Siamese Bi-LSTM networks to fuse the enhanced representations of answer texts and combine their multiple pooled vectors for grade classification on a multi-linear prediction layer.

The remainder of this paper is organized as follows: Section II summarizes and details related works. Section III defines the ASAG task and proposes a cross-lingual hybrid neural network with interaction enhancement for ASAG. Section IV describes the evaluated datasets and experimental settings and presents the experimental results. Finally, Section V concludes the paper.

II. RELATED STUDIES

In the early stages, deep learning was usually involved in ASAG tasks only as a complement to feature-based methods, where deep learning provides only some features, but not all, for the ASAG task. For example, Marvaniya et al. [5] and Saha et al. [2] used a pre-trained Bi-LSTM network InferSent [19] to obtain sentence embedding vectors of answer texts, which compensates for the disadvantage of not being able to represent the context in token overlap methods. Tan et al. [20] proposed a scoring method using a combination of Graph Convolutional Networks (GCNs) with several sparse features. First, they used word/bigram-level nodes, sentence-level nodes and edges between nodes to construct an undirected heterogeneous text graph for the answer text. They then proposed a two-layer GCN model to encode the graph representation of the answer text. Moreover, Zhang et al. [21] used a Deep Belief Network (DBN), whose input is six sparse features, instead of traditional machine learning to classify student answers.

To automatically extract features from the answer text in an end-to-end manner, deep learning methods that independently undertake ASAG tasks have begun to appear. For example, Kumar et al. [13] proposed a Bi-LSTM-based model for the ASAG task. They used two Siamese Bi-LSTMs to encode contexts for the reference answer and student answer, used earth-mover distance (EMD) pooling to interact hidden states from two Bi-LSTMs, and used a flexible regression layer to predict scores. Uto and Uchida [14] introduced item response theory into the LSTM network for grading short answers. Tulu et al. [15] added sense vectors and used Manhattan distance pooling to improve the LSTM-based ASAG method [13], [14]. Riordan et al. [16] combined an LSTM network with a low-level CNN for

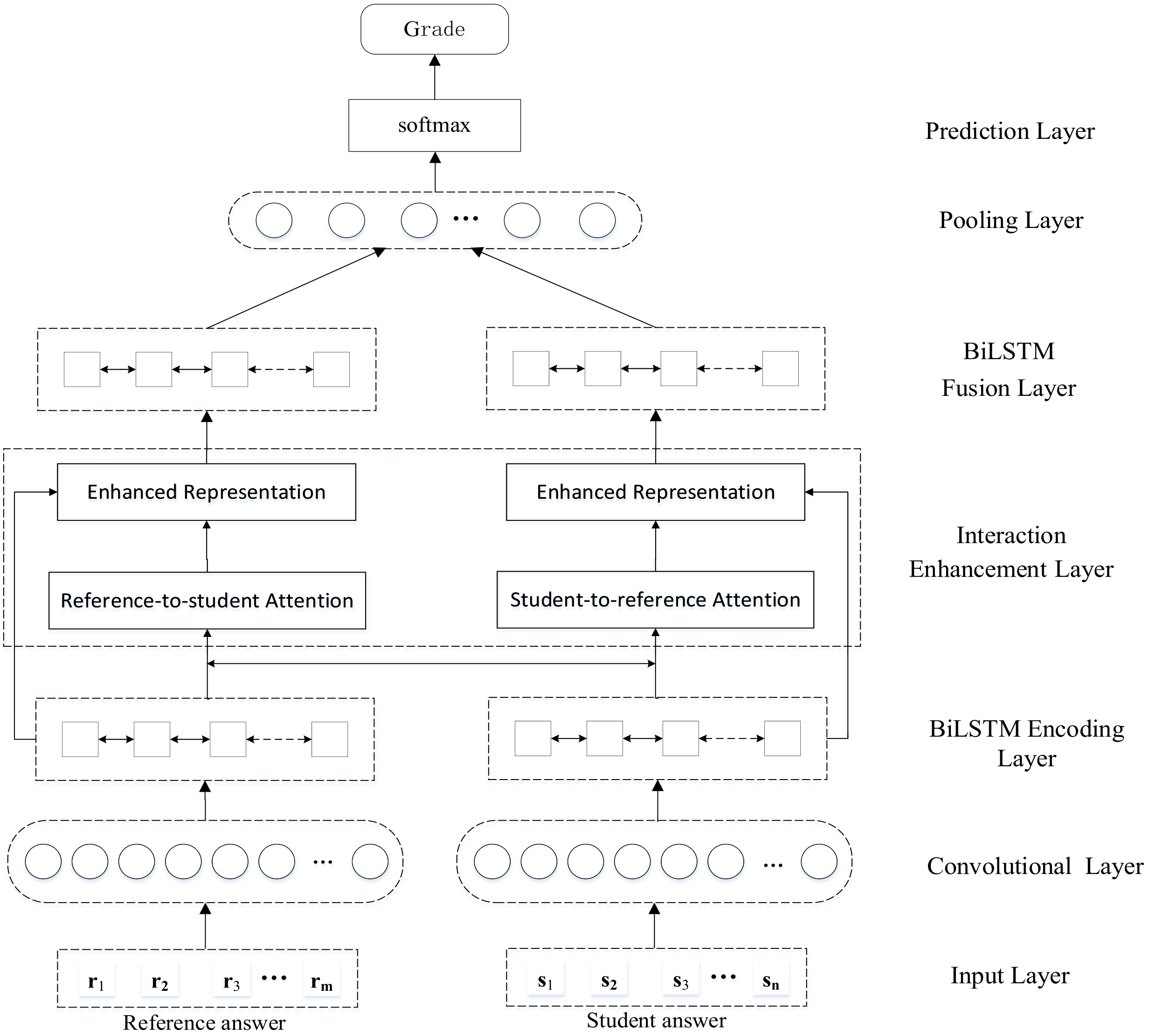

FIGURE 1. Proposed hybrid neural network model framework for ASAG.

ASAG. Liu et al. [17] proposed a Transformer -based multiway attention model [22] for ASAG using a large K-12 dataset.

The aforementioned deep learning methods [13], [14], [15], [16], [17] achieve feature extraction and scoring in an end-to-end manner, but their coding structure is relatively singular and simple, and they cannot extract comprehensive and effective features from answer texts. To solve this problem, this study used hybrid neural networks to propose a cross-lingual deep learning model for grading short- answer texts.

III. METHODOLOGY

A. TASK DEFINITION

The goal of the ASAG task is to assign a score value to a given student's answer based on the reference answer, which is essentially a regression task. To facilitate the processing of the ASAG task in a neural network classification model, we space the answer scores to form score interval categories. For example, for the dataset with scores between 0 and 5, as shown in Table 1, we set 11 score intervals: [0, 0], (0, 0.5], (0.5, 1], …, (4.5, 5].

Based on the mapping of the above score intervals, we convert the ASAG task into a multi-classification task. Our goal is to train a model $P(\cdot;\theta)$ for the ASAG task to predict the score interval y^* for the specified answer pair (R, S) as follows:

$$y^* = \underset{y\in Y}{argmax}(P(y|(R,S);\theta)) \tag{1}$$

where Y denotes the set of score intervals.

B. PROPOSED HYBRID NEURAL NETWORK MODEL

The proposed hybrid neural network model consists of multiple layers of Siamese neural networks with attention mechanisms. The framework structure of our model is shown in Figure 1, and it has the following characteristics:

(1) In terms of encoding, we first use two Siamese CNNs that share parameters to extract phrase features in answer texts to form meaningful linguistic segments. This process is equivalent to word segmentation for Chinese text, and word segmentation is very important for Chinese because meaningful language fragments in Chinese are usually phrases, rather than single words. We then use two Siamese Bi-LSTM networks to encode the contextual information of the answer texts from the local features.

(2) In terms of information interaction, we use the dot-product attention mechanism [22] to form the reference-answer-to-student-answer and student-answer-to-reference-answer attentions, respectively. More importantly, we combine the input and output of attention through various forms, such as concatenation, difference, and product, to form the enhanced representation of the answer text.

(3) Finally, we fuse the enhanced representations of answer texts using two Siamese Bi-LSTM networks to obtain the final semantic representations of answer texts and combine their multiple pooled vectors for grade classification.

C. INPUT LAYER

To provide additional semantic information for grading, we concatenate questions and answers to form unique answer input sequences. First, we convert the words in the question, reference answer, and student answer to the corresponding word embedding vectors to obtain the question sequence $T \in \mathbb{R}^{d_w \times e}$, reference answer sequence $P \in \mathbb{R}^{d_w \times u}$, and student answer sequence $Q \in \mathbb{R}^{d_w \times v}$. Then, we connect the question sequence with the reference and student answer sequences to obtain the reference answer input R and student answer input S, respectively. The calculation process is as follows:

$$R = [T; P] = \{r_1, r_2, \ldots, r_m\} \in \mathbb{R}^{d_w \times m} \tag{2}$$

$$S = [T; Q] = \{s_1, s_2, \ldots, r_n\} \in \mathbb{R}^{d_w \times n} \tag{3}$$

where d_w represents the dimension of the word embedding vector, $m = e + u$, $n = e + v$, and e, u, and v represent the numbers of words in the question, reference answer and student answer, respectively.

D. CONVOLUTIONAL LAYER

We use two Siamese CNNs to extract local features from the input sequences to form phrases in the answer text. This is especially important for Chinese answers as it is equivalent to the word segmentation process of Chinese text. We use zero padding in the convolutional layer to ensure that its output is a sequence of the same length as the input. The calculation process is as follows:

$$R_c = ReLU(CNN\,(R)) \in \mathbb{R}^{d_C \times m} \tag{4}$$

$$S_c = ReLU(CNN\,(S)) \in \mathbb{R}^{d_C \times n} \tag{5}$$

where $CNN\,(\cdot)$ denotes the Siamese CNN with zero padding and shared parameters, d_c is the number of convolution kernels, and $ReLU(\cdot)$ denotes the Rectified Linear Unit activation function.

E. BI-LSTM ENCODING LAYER

After obtaining local features, we further use two Siamese Bi-LSTM networks to encode the contextual information of answer texts from the local features as follows:

$$R_e = \left[\overrightarrow{LSTM1}\,(R_c)\,; \overleftarrow{LSTM1}\,(R_c)\right] = \{r_i^e \in \mathbb{R}^{2d_h}\}_{i=1}^{m} \tag{6}$$

$$S_e = [\overrightarrow{LSTM1}\,(S_c)\,; \overleftarrow{LSTM1}\,(S_c)] = \{s_j^e \in \mathbb{R}^{2d_h}\}_{j=1}^{n} \tag{7}$$

where $\overrightarrow{LSTM1}\,(\cdot)$ and $\overleftarrow{LSTM1}\,(\cdot)$ denote left-to-right and right-to-left LSTM networks in the Siamese Bi-LSTM encoding network, respectively, in which the number of hidden units for both $\overrightarrow{LSTM1}\,(\cdot)$ and $\overleftarrow{LSTM1}\,(\cdot)$ is d_h. ";" indicates the connection operation in the hidden state, and r_i^e and s_j^e are the hidden states in R_e and S_e, respectively.

F. INTERACTION ENHANCEMENT LAYER

We first compute the interaction information using the following dot-product attention steps:

(1) We use the dot product similarity between elements in the hidden state tuple $< r_i^e, s_j^e >$ to represent the attention weights between them as follows:

$$\alpha_{ij} = \left(r_i^e\right)^T s_j^e \tag{8}$$

(2) We compute the attention encodings $\widetilde{R}_e = \{\widetilde{r}_i^e\}_{i=1}^{m}$ and $\widetilde{S}_e = \{\widetilde{s}_j^e\}_{j=1}^{n}$ respectively using the following weight sums:

$$\widetilde{r}_i^e = \sum_{j=1}^{n} \frac{\exp(\alpha_{ij})}{\sum_{k=1}^{n} \exp(\alpha_{ik})} s_j^e \in \mathbb{R}^{2d_h}, \quad i = 1, 2, \ldots, m \tag{9}$$

$$\widetilde{s}_j^e = \sum_{i=1}^{m} \frac{\exp(\alpha_{ij})}{\sum_{k=1}^{m} \exp(\alpha_{kj})} r_i^e \in \mathbb{R}^{2d_h}, \quad j = 1, 2, \ldots, n \tag{10}$$

Based on the attention encodings above, we obtain the enhanced representations $R_a = \{r_i^a\}_{i=1}^{m}$ and $S_a = \{s_j^a\}_{j=1}^{n}$ for the answer texts R and S, where $r_i^a \in \mathbb{R}^{8d_h}$ and $s_j^a \in \mathbb{R}^{8d_h}$ are computed through various combinations of their attention and context as follows:

$$r_i^a = [r_i^e; \widetilde{r}_i^e; r_i^e - \widetilde{r}_i^e; r_i^e \odot \widetilde{r}_i^e] \tag{11}$$

$$s_j^a = [s_j^e; \widetilde{s}_j^e; s_j^e - \widetilde{s}_j^e; s_j^e \odot \widetilde{s}_j^e] \tag{12}$$

where $\odot$ denotes element-wise multiplication.

G. BI-LSTM FUSION LAYER

For the enhanced representations of the answer texts, we further use two Siamese Bi-LSTMs to fuse their various combinations, which are computed as follows:

$$R_f = [\overrightarrow{LSTM2}\,(R_a)\, \overleftarrow{LSTM2}\,(R_a)] \in \mathbb{R}^{2d_f \times m} \tag{13}$$

$$S_f = [\overrightarrow{LSTM2}\,(S_a)\,; \overleftarrow{LSTM2}\,(S_a)] \in \mathbb{R}^{2d_f \times n} \tag{14}$$

where $\overrightarrow{LSTM2}\,(\cdot)$ and $\overleftarrow{LSTM2}\,(\cdot)$ denote left-to-right and right-to-left LSTM networks in the Siamese Bi-LSTM fusion network, respectively; and the number of hidden units for both $\overrightarrow{LSTM2}(\cdot)$ and $\overleftarrow{LSTM2}(\cdot)$ is d_f.

H. POOLING LAYER

We comprehensively consider max pooling and average pooling to form a classification vector Z from the fusion semantics of the answer text. The calculation process is as follows:

$$Z^r_{max} = maxPool(R_f) \in \mathbb{R}^{2d_f} \tag{15}$$

$$Z^r_{ave} = avePool(R_f) \in \mathbb{R}^{2d_f} \tag{16}$$

$$Z^s_{max} = maxPool(S_f) \in \mathbb{R}^{2d_f} \tag{17}$$

$$Z^s_{ave} = avePool(S_f) \in \mathbb{R}^{2d_f} \tag{18}$$

$$Z = \left[Z^r_{max}; Z^r_{ave}; Z^s_{max}; Z^s_{ave}\right] \in \mathbb{R}^{8d_f} \tag{19}$$

where $maxPool(\cdot)$ and $avePool(\cdot)$ denote the max-pooling and average-pooling operations, respectively.

I. PREDICTION LAYER

We use two fully-connected layers to convert the classification vector to the probability of the predicted score interval, which are computed as follows:

$$O^1 = drop(ReLU(W^1_o Z + b^1_o)) \in \mathbb{R}^{d_p} \tag{20}$$

$$O^2 = W^2_o O^1 + b^2_o \in \mathbb{R}^{d_y} \tag{21}$$

$$P\left(\hat{y}\,|\,Z;\theta\right) = \frac{\exp(O^2_{\hat{y}})}{\sum_{y\in Y}\exp(O^2_y)} \tag{22}$$

where $W^1_o \in \mathbb{R}^{d_p\times 8d_f}$ and $b^1_o \in \mathbb{R}^{d_p}$ are the weight and bias of the first fully-connected layer, respectively; d_p is the number of hidden units in the prediction layer; $drop(\cdot)$ denotes the dropout operation [23]; O^1 is the output of the first fully-connected layer; $W^2_o \in \mathbb{R}^{d_y\times d_p}$ and $b^2_o \in \mathbb{R}^{d_y}$ are the weight and bias of the second fully-connected layer, respectively; d_y is the number of score intervals; O^2 is the confidence vector for various rating intervals; $\hat{y}$ is the predicted score interval; and θ denotes all parameters of the model.

J. MODEL TRAINING

We use the cross-entropy loss function to calculate the error between annotation and prediction during training. Let Ω be the set of answer pairs in a training batch, y_i be the ground-truth score interval of the i-th answer pair in Ω, and Z_i be the classification vector of the i-th score interval in Ω. The goal of the training is to minimize the following loss error for all answer pairs in Ω:

$$L(\theta) = -\frac{1}{|\Omega|}\sum_{i=1}^{|\Omega|}\log(P(y_i|Z_i,\theta)) \tag{23}$$

IV. EXPERIMENTS

A. DATASETS

To evaluate the performance of our model in different languages, we used two ASAG task datasets in English and Chinese, including the widely used English Mohler dataset [3] for the ASAG problem and a Chinese *Computer Network* ASAG dataset that we built ourselves, as shown in Table 2.

TABLE 2. Details of two datasets used for evaluation.

		Total	Training set
English dataset	Mohler	30,000	Randomly select in 12-fold cross validation
Chinese Network	Computer	7896	6318

TABLE 3. Hyper-parameters settings in our model.

Parameter	Setting	Explanation
d_w	300	Dimension of word embedding
d_c	150	Number of convolution kernels
d_h	100	Number of hidden units in LSTM1
d_f	200	Number of hidden units in LSTM2
d_p	1024	Number of hidden units in prediction
epochs	20	Number of iterations
batch	64	Batch size
lr	1e-4	Initial learning rate
dropout	0.1	Random dropout rate

English Mohler dataset [3] is an English ASAG dataset created by Mohler et al. at the University of North Texas, which includes 2,273 student answers to 80 questions on introductory computer science. Each student's answer was graded by two teachers using an integer from 0 to 5, where we take their average as the true label of the student answer. This dataset has been widely evaluated in several studies [2], [3], [4], [7], [13].

There are only 2273 answer pairs in the Mohler dataset, which is too few for our hybrid neural network model. To address this problem, we extended the dataset using the method proposed by Kumar et al. [12], where they took the correct student answers in the training set as additional reference answers and extended the number of training answer pairs to approximately 30,000. We used the Mohler dataset in 12-fold cross-validation to evaluate our model.

Chinese Computer Network dataset is a Chinese ASAG dataset built for this study. It contains 7896 answer pairs to various questions about computer networking, 6318 for training, 789 for validation, and 789 for test. Each student's answer was graded by three teachers, using an integer from 0 to 10.

B. EXPERIMENTAL SETTINGS

The English word embedding vector used in our model is a 300-dimensional Glove vector [24], and the Chinese word embedding vectors used in our model is 300-dimensional Word2vec pre-trained by Liu,[1] which was obtained by training on offline dumps of Chinese Wikipedia. The number of convolution kernels in the convolutional layer was set to 150, and the sliding window size was set to 3. The number of hidden units of each LSTM network in the Bi-LSTM encoding layer was set to 100, and the number of hidden units in the

[1] https://github.com/liuhuanyong/ChineseEmbedding

TABLE 4. Experiments on english mohler dataset.

System/Model	MAE	RMSE	Pearson r	Acc	F1
LSTM-Last	0.91+	1.101+	0.600+	0.784	0.651
LSTM-Max	1.12+	1.60+	0.411+	0.609	0.484
LSTM-Avg	1.16+	1.58+	0.393+	0.586	0.432
LSTM-EMD	0.657+	1.135+	0.649+	0.834	0.712
LSTM+CNN	0.712	2.418	0.669	0.892	0.766
CNN	0.232	0.933	0.905	0.913	0.816
Bi-LSTM+ Interaction	0.083	0.546	0.955	0.951	0.956
Our model+	**0.064**	**0.524**	**0.970**	**0.977**	**0.961**

Lower is better for MAE and RMSE; higher is better for Accuracy and Pearson's r. The results with symbol "+" are retrieved from [13] and others are from this work.

TABLE 5. Experiments on chinese computer network dataset.

System/Model	MAE	RMSE	Pearson r	Acc	F1
LSTM-Last++	1.61	1.85	0.231	0.306	0.185
LSTM-Max++	1.72	1.92	0.217	0.298	0.182
LSTM-Avg++	1.66	1.88	0.205	0.295	0.18
LSTM-EMD++	1.52	1.75	0.288	0.312	0.186
LSTM+CNN	1.55	1.949	0.202	0.328	0.188
CNN	1.59	1.988	0.266	0.280	0.170
Bi-LSTM+ Interaction	0.662	1.128	0.572	0.596	0.410
Our model+	**0.429**	**1.048**	**0.606**	**0.656**	**0.492**

Bi-LSTM fusion layer was set to 200. The hyper-parameters of our model are listed in Table 3. Software running platform is TensorFlow-gpu 2.2.0.

C. COMPARISON WITH BASELINE MODELS

We compared our model with the following baseline models for ASAG:

LSTM-EMD [13] is a deep learning method that uses only Bi-LSTM and a pooling layer based on the earth-mover distance. We also quote its three results: LSTM-Last using only the last hidden state, LSTM-Max using maximum pooling, and LSTM-Avg using average pooling.

CNN + Bi-LSTM is a CNN and Bi-LSTM to form a deep learning model without an interaction layer. It is implemented in this study.

Bi-LSTM + Interaction is a simplified version of our model after removing CNN.

CNN is a simplified version of our model only using CNN.

D. EXPERIMENTAL RESULTS AND DISCUSSIONS

Table 4 presents the experimental results of the various models on the English Mohler dataset. The experimental results shows that our hybrid neural network with interaction enhancement significantly improves the performance of various simple models, such as single CNN or LSTM, on the English Mohler dataset. The experimental results in Table 4 also show that the performance of our model does not drop significantly after removing the CNN; however, the model performance drops sharply after removing the interaction enhancement layer, which reveals that interaction enhancement plays a very important role in the English ASAG task. In addition, the experimental results show that the performance of a single CNN model is significantly better than that of the LSTM model in the English ASAG task.

Table 5 presents the experimental results of the various models on the Chinese Computer Network dataset. The experimental results show that our hybrid neural network with interaction enhancement significantly improves the performance of various simple models, such as single CNN or LSTM, in the Chinese ASAG task. The experimental results in Table 5 also show that, similar to the English ASAG task, interaction enhancement plays a very important role in the Chinese ASAG task. In addition, the results in Table 5 show that the overall performance of the model in the Chinese Computer Network dataset is significantly lower than that in the English Mohler dataset. We believe that this is due to the fact that the Chinese Computer Network dataset has not been extended and is smaller than the English Mohler dataset.

V. CONCLUSION

In this paper, we proposed a novel cross-lingual hybrid neural networks with interaction enhancement for ASAG that consists of a convolutional layer, Bi-LSTM encoding layer, interaction enhancement layer, Bi-LSTM fusion layer, pooling layer, and prediction layer. Through experimental evaluation, this study revealed the following: (1) Simple models, such as single CNN or LSTM, on the English Mohler dataset, cannot play an important role in both English and Chinese ASAG tasks; (2) interaction enhancement plays a very important role in both English and Chinese ASAG tasks; and (3) in the front end of Bi-LSTM, a CNN that extracts phrase information can improve the scoring accuracy of the model.

REFERENCES

[1] G. G. Smith, R. Haworth, and S. Žitnik, "Computer science meets education: Natural language processing for automatic grading of open-ended questions in eBooks," *J. Educ. Comput. Res.*, vol. 58, no. 7, pp. 1227–1255, Dec. 2020.

[2] S. Saha, T. I. Dhamecha, S. Marvaniya, R. Sindhgatta, and B. Sengupta, "Sentence level or token level features for automatic short answer grading: Use both," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2018, pp. 503–517.

[3] M. Mohler, R. Bunescu, and R. Mihalcea, "Learning to grade short answer questions using semantic similarity measures and dependency graph alignments," in *Proc. ACL*, 2011, pp. 752–762.

[4] M. A. Sultan, C. Salazar, and T. Sumner, "Fast and easy short answer grading with high accuracy," in *Proc. NAACL-HLT*, 2016, pp. 1070–1075.

[5] S. Marvaniya, S. Saha, T. I. Dhamecha, P. Foltz, R. Sindhgatta, and B. Sengupta, "Creating scoring rubric from representative student answers for improved short answer grading," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 993–1002.

[6] A. Sahu and P. K. Bhowmick, "Feature engineering and ensemble-based approach for improving automatic short-answer grading performance," *IEEE Trans. Learn. Technol.*, vol. 13, no. 1, pp. 77–90, Jan. 2020.

[7] S. Roy, H. S. Bhatt, and Y. Narahari, "An iterative transfer learning based ensemble technique for automatic short answer grading," 2016, *arXiv:1609.04909*.

[8] N. Süzen, A. N. Gorban, J. Levesley, and E. M. Mirkes, "Automatic short answer grading and feedback using text mining methods," *Proc. Comput. Sci.*, vol. 169, pp. 726–743, Jan. 2020.

[9] M. Heilman and N. Madnani, "ETS: Discriminative edit models for paraphrase scoring," in *Proc. 1st Joint Conf. Lexical Comput. Semantics (SEM)*, 2012, pp. 529–535.

[10] S. Jimenez, C. Becerra, and A. Gelbukh, "SOFTCARDINALITY: Hierarchical text overlap for student response analysis," in *Proc. Joint Conf. Lexical Comput. Semantics*, vol. 2, 2013, pp. 280–284.

[11] G. Jorge-Botana, J. M. Luzón, I. Gómez-Veiga, and J. I. Martín-Cordero, "Automated LSA assessment of summaries in distance education: Some variables to be considered," *J. Educ. Comput. Res.*, vol. 52, no. 3, pp. 341–364, Jun. 2015.

[12] D. Alikaniotis, H. Yannakoudakis, and M. Rei, "Automatic text scoring using neural networks," in *Proc. ACL*, vol. 1, 2016, pp. 715–725.

[13] S. Kumar, S. Chakrabarti, and S. Roy, "Earth mover's distance pooling over Siamese LSTMs for automatic short answer grading," in *Proc. Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2046–2052.

[14] M. Uto and Y. Uchida, "Automated short-answer grading using deep neural networks and item response theory," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2020, pp. 334–339.

[15] C. N. Tulu, O. Ozkaya, and U. Orhan, "Automatic short answer grading with SemSpace sense vectors and MaLSTM," *IEEE Access*, vol. 9, pp. 19270–19280, 2021.

[16] B. Riordan, A. Horbach, A. Cahill, T. Zesch, and C. M. Lee, "Investigating neural architectures for short answer scoring," in *Proc. 12th Workshop Innov. Use NLP Building Educ. Appl.*, 2017, pp. 159–168.

[17] T. Liu, W. Ding, Z. Wang, J. Tang, G. Huang, and Z. Liu, "Automatic short answer grading via multiway attention networks," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2019, pp. 169–173.

[18] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced LSTM for natural language inference," in *Proc. ACL*, 2017, pp. 1657–1668.

[19] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 670–680.

[20] H. Tan, C. Wang, Q. Duan, Y. Lu, H. Zhang, and R. Li, "Automatic short answer grading by encoding student responses via a graph convolutional network," *Interact. Learn. Environ.*, vol. 2020, no. 2, pp. 1–15, Dec. 2020.

[21] Y. Zhang, C. Lin, and M. Chi, "Going deeper: Automatic short-answer grading by combining student and question models," *User Model. User-Adapted Interact.*, vol. 30, no. 1, pp. 51–80, Mar. 2020.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.

YISHAN CHEN received the B.S. degree in computer and application from the Central South Institute of Technology, Hengyang, China, in 1998, and the master's degree in computer software and theory from Guangxi Normal University.

His research interests include intelligent teaching systems and natural language processing.

JIANHUA LUO received the B.S. degree in mathematics from Guangxi University, Guangxi, China, in 1987, and the M.S. degree in computer software from Guangxi Normal University, Guangxi, in 1996.

He is a Professor with the School of Business, Guilin Tourism University, China. His research interests include natural language processing and intelligent tutoring systems.

XINHUA ZHU received the B.S. degree in computer science from the Department of Radio Electronics, Beijing Normal University, China.

He is a Professor with the School of Computer Science and Engineering, Guangxi Normal University, China. His research interests include intelligent tutoring systems, natural language processing, and knowledge graphs.

HAN WU received the bachelor's degree in Internet of Things engineering from Yancheng Normal University, China, in 1989. He is currently pursuing the master's degree with the School of Computer Science and Engineering, Guangxi Normal University, China. His research interests include intelligent tutoring systems and neural computing.

SHANGBO YUAN received the bachelor's degree in energy and power engineering from Shandong University, China, in 2021. He is currently pursuing the master's degree with the School of Engineering and Design, Technical University of Munich, Germany. His research interests include neural computing and energy systems.