# Automatic Short-Answer Grading via BERT-Based Deep Neural Networks

Xinhua Zhu, Han Wu, and Lanfang Zhang

***Abstract*—Automatic short-answer grading (ASAG) is a key component of intelligent tutoring systems. Deep learning is an advanced method to deal with recognizing textual entailment tasks in an end-to-end manner. However, deep learning methods for ASAG still remain challenging mainly because of the following two major reasons: 1) high-precision scoring requires a deep understanding of the answer text; and 2) ASAG's corpus is usually small and cannot provide enough training data for deep learning. To address these challenges, in this article, we propose a novel bidirectional encoder representation from transformer (BERT)-based deep neural network framework for ASAG. First, we use a pretrained and fine-tuned BERT model to dynamically encode the answer text, which can effectively overcome the problem of a too small corpus in the ASAG task. Second, to generate a powerful semantic representation for ASAG, we construct a semantic refinement layer to refine the semantics of the BERT outputs, which consists of a bidirectional-Long Short-Term Memory (LSTM) network and a Capsule network with position information in parallel. Third, we propose a triple-hot loss strategy for regression tasks in ASAG, which changes the gold label representation in the standard cross-entropy loss function from one-hot to triple-hot. Experiments demonstrate that our proposed model is effective and outperforms most of the state-of-the-art systems on both the SemEval-2013 dataset and the Mohler dataset. The code is available online at https://github.com/wuhan-1222/ASAG.**

***Index Terms*—Automatic short-answer grading (ASAG), BERT language model, Bi-LSTM network, Capsule network, intelligent tutoring systems (ITSs), textual entailment.**

## I. Introduction

ADAPTIVE test and assessment are the key components of intelligent tutoring systems (ITSs) [39], which can capture the current cognitive level of students and provide a basis for the system to formulate a personalized learning route. For the simplicity of implementation, standardized multiple-choice or matching questions are widely used in adaptive assessments [37], [39]. However, standardized questions have the following two obvious shortcomings: 1) standardized questions usually only provide a few alternative items and cannot list all possible students' responses; and 2) students may randomly choose answers from the listed items without thinking, thus not accurately capturing their cognitive level. Automatic short-answer grading (ASAG) can effectively overcome the above-mentioned shortcomings by requiring students to actively answer the questions using short text and automatic grading answer texts [17], [36]. More importantly, ASAG can directly associate students' answers with cognitive types, such as *Correct, Partially correct, Contradictory, Irrelevant, and Nondomain* [17]. Therefore, ASAG is increasingly becoming an important approach to adaptive assessment in ITSs. Since both students' answers and reference answers appear in the form of natural language, ASAG is regarded as an application of recognizing textual entailment in educational technology [1], [17].

Feature engineering is the dominant approach in most previous studies [1]–[9] on ASAG. Various sparse features are investigated and utilized for ASAG. For example, token overlap features [1], [5], [6], [9], syntax and dependency features [2], [4], knowledge-based features using WordNet [5], [6], and text features using tf-idf [3], [6], [7], [10], edits [8] or sentence embedding [1], [4], [6]. However, there are still some problems in the feature engineering-based methods. First, sparse feature extractions require many preprocessing steps, such as lemmatization, token boundary, part-of-speech tagging, and dependency and constituency parses, whereas each preprocessing step may cause certain errors, which would result in error transmission and accumulation. On the other hand, feature engineering lacks effective methods to encode the text sequence and cannot effectively construct the context information for the answer text.

With the continuous development of artificial neural network technologies, many deep learning models, such as Long Short-Term Memory (LSTM)-based models [11]–[14], convolutional neural network (CNN)- and LSTM-based model [15], and transformer-based model [16], are applied to short-answer grading. These deep learning models utilize different neural networks to automatically extract local and global semantic information from the answer text converted into a word embedding sequence, thereby providing an end-to-end approach without any feature engineering effort [16]. However, deep learning methods for ASAG still remain challenging mainly for two reasons. First, students usually use different free texts to answer the same question, in which

Manuscript received 19 July 2021; revised 11 April 2022; accepted 10 May 2022. Date of publication 19 May 2022; date of current version 5 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62062012 and Grant 61967003, in part by the Natural Science Foundation of Guangxi of China under Grant 2020GXNSFAA159082, and in part by the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing. *(Corresponding author: Lanfang Zhang.)*

Xinhua Zhu is with the Guangxi Key Lab of Multi-Source Information Mining and Security and the School of Computer Science and Engineering, Guangxi Normal University, Guilin 541004, China (e-mail: zxh429@263.net).

Han Wu is with the School of Computer Science and Engineering, Guangxi Normal University, Guilin 541004, China (e-mail: 1402831802@qq.com).

Lanfang Zhang is with the Guangxi Key Lab of Multi-Source Information Mining and Security and the Faculty of Education, Guangxi Normal University, Guilin 541004, China (e-mail: 814611301@qq.com).

Digital Object Identifier 10.1109/TLT.2022.3175537

TABLE I
EXAMPLE OF LABELED STUDENTS' ANSWERS IN THE SEMEVAL-2013 DATASET [17]

| | |
|---|---|
| QUESTION | Carrie wanted to find out which was harder, a penny or a nickel, so she did a scratch test. How would this tell her which is harder? |
| REF. ANS. | The harder coin will scratch the other. |
| STUD. ANS. | (1) If just the penny could scratch and the nickel could not the penny is harder. **(accuracy="correct")** |
| | (2) Which had less scratches. **(accuracy="correct")** |
| | (3) A paperclip is harder. **(accuracy="irrelevant")** |
| | (4) I do not know. **(accuracy="non_domain")** |

students' answers may have significant differences in sentence structure, language style, and text length. Therefore, it is necessary to utilize advanced learning techniques to combine different deep neural networks in the ASAG task in order to achieve a deeper semantic understanding of students' answers. Second, the deep learning method for ASAG is fully supervised machine learning, which requires assigning a label to each students' answer in the training corpus, as shown in Table I. Since it is very time-consuming to accurately assign a label to a freely expressed students' answer, the training corpus for ASAG is usually small in size, which is usually only a few thousand instead of the tens of thousands in conventional deep learning tasks [41]. Therefore, how to train a stable and effective deep neural network model on a small corpus is a major challenge faced by the deep learning methods of ASAG.

In this article, to address the issues and challenges mentioned above, we take full advantage of the semantic complementarity and mutual promotion between the bidirectional encoder representation from transformer (BERT) model and the classical neural network and propose a BERT-based deep neural network framework to tackle the ASAG problem in a more effective way. The main contributions of this article can be summarized as follows.

1) We extend the application of the pretrained BERT model [18] in ASAG tasks from a fine-tuning approach [19], [20] to a combination with bidirectional LSTM (Bi-LSTM) and Capsule [21] networks and experimentally demonstrate that this combination is effective and outperforms most of the state-of-the-art systems on both the Semantic Evaluation (SemEval) Workshop in 2013 dataset (SemEval-2013 dataset) [17] and the Mohler dataset [2].
2) On top of the fine-tuned BERT model, we construct a semantic refinement layer to refine the semantics of BERT outputs, which consists of a Bi-LSTM network and a Capsule network with position information. Specifically, we employ the complex gate structures in the Bi-LSTM network to extract fine global context for the BERT outputs and employ convolutional capsules in the Capsule network to extract the related local context for the hidden states of the BERT model.
3) We fuse the fine global context and local context by using a multihead attention [25] to generate a powerful semantic representation for ASAG.
4) We propose a triple-hot loss strategy for regression tasks in ASAG, which changes the gold label representation in the standard cross-entropy loss function from one-hot to triple-hot, and a compromise method to double the training set in the Mohler dataset, which only selects a correct students' answer from each question as a supplementary reference answer.

The rest of this article is organized as follows. Section II summarizes and details the related works. Section III defines the task category in the ASAG problem and proposes a BERT-based deep neural network framework for ASAG. Section IV describes evaluated datasets and experimental settings and shows experimental results. Section V gives some discussions about the experimental results. Finally, Section VI concludes our work and outlines future directions.

## II. RELATED WORK

### A. Applications of Deep Learning in ASAG Tasks

According to the role of deep learning in ASAG tasks and its training approach, the application of deep learning in ASAG tasks can be divided into the following three types.

1) Participator: Deep learning participates in the feature-based methods, in which deep learning undertakes subtasks at a certain phase or only calculates part of the features of the answer text.
2) Contractor: Deep learning independently undertakes the ASAG task in an end-to-end manner.
3) Transfer learning methods that use a deep neural network pretrained on large-scale corpora to grade the short answer.

The development process of each method is described in detail as follows.

Combining deep learning with sparse features is an important grading method for ASAG. For example, Marvaniya *et al.* [4] and Saha *et al.* [1] used a pretrained neural network InferSent [22] to encode answer texts, which makes up for the lack of contextual representation in token overlap methods, in which InferSent is a pretrained sentence embedding model using a Bi-LSTM network. Tan *et al.* [23] proposed a scoring method that combines graph convolutional networks (GCNs) with several sparse features. They first constructed an undirected heterogeneous text graph for answer texts with the sentence-level nodes, the word/bigram-level nodes, and the edges between nodes. Then, they used a two-layer GCN model to encode the graph structure and obtain the graph representation. Moreover, Zhang *et al.* [24] utilized a deep belief network as a classifier instead of traditional machine learning to classify students' answer representations that are composed of six sparse features.

To automatically learn useful features from answer texts, Kumar *et al.* [12] proposed a Bi-LSTM framework for ASAG. Their framework consists of three cascaded neural modules: Siamese Bi-LSTMs, respectively, applied to a reference and a student answer, a pooling layer that uses earth-mover distance (EMD) to interact hidden states from both LSTMs, and a flexible regression layer to output scores. Uto and Uchida [13] combined the LSTM network with item response theory for short-answer grading. Tulu *et al.* [14] improved the LSTM-based grading

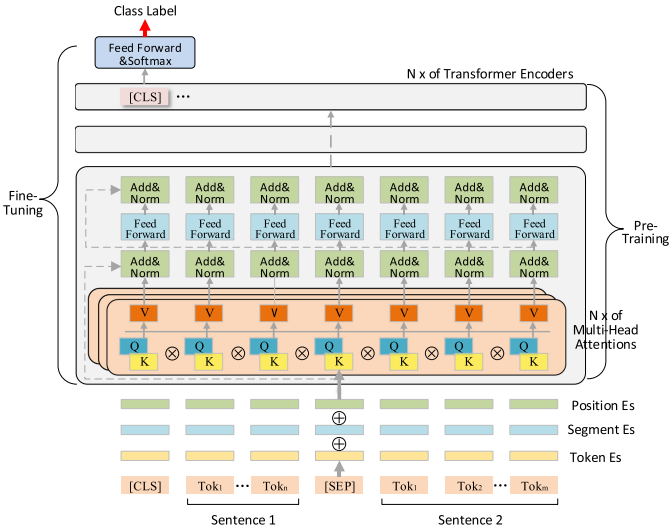


Fig. 1. BERT model architecture. We employed the sentence pair classification task specific model to describe the details of the BERT model.

method [12], [13] by introducing sense vectors and replacing the pooling layer with Manhattan distance. Riordan *et al.* [15] combined CNN and the LSTM network for short-answer grading. Liu *et al.* [16] proposed a transformer [25] model with multiway attention for ASAG on a large K-12 dataset. Transformer is a new type of neural network encoder based solely on attention mechanisms and can train efficiently on large corpora in a parallel manner.

The above-mentioned deep learning methods [12]–[16] achieve grading and scoring in an end-to-end manner, but they require a large amount of labeled corpus for training their model, which is what most ASAG corpora lack. To solve this problem, various pretrained transfer learning models [26], such as embeddings from language models (ELMo) [27], BERT [18], generative pretrained transformer (GPT) [28], and GPT-2 [29], are applied to the ASAG task. Among them, BERT is especially outstanding and has achieved state-of-the-art grading results [19], [20].

### B. BERT Model and Its Application in Education

BERT is a new pretrained language representation model, which is developed by Google AI Language on the basis of absorbing the advantages of ELMo and GPT. On the one hand, as GPT does, BERT uses a stacked transformer [25] architecture to train the weights for transfer learning, which provides more structured memory for the long-term dependencies compared to the LSTM networks in ELMo. On the other hand, it uses a left-to-right and a right-to-left language model in ELMo instead of the unidirectional language model in GPT to capture bidirectional context. The BERT model architecture is shown in Fig. 1 [20].

To adapt to various downstream tasks, BERT employs one token sequence to unambiguously represent the inputs of both a single sentence and a pair of sentences. The first token of each sequence is always a special classification token (CLS) whose final hidden state is used as the aggregate sequence representation of the classification task. To separate a sentence pair in the input sequence, BERT inserts a special token (SEP) between two sentences and adds a learned segment embedding to each token representation to indicate which sentence it belongs to. Finally, the input embedding of each token in the sequence is the sum of the token embedding, the segmentation embedding, and the position embedding, as shown in Fig. 1.

BERT model architecture is a multilayer bidirectional transformer encoder [25]. The transformer in BERT uses a multihead attention to jointly attend to information from different representation subspaces at different positions in the input sequence. Each layer of the transformer also contains a fully connected feed-forward network behind the multihead attention. Owing to the portability of the transformer encoder, the BERT model stacks 12 layers of transformers in its basic version $\text{BERT}_{\text{BASE}}$, which is far more than the three-layer structure in the ELMo model. Therefore, BERT can capture deeper semantics and create powerful sequence representations that perform extremely well on many downstream tasks [18].

The BERT model is pretrained by two unsupervised tasks of masked language model and next sentence prediction in a huge corpus that consists of a Book Corpus with 800M words and an English Wikipedia with 2500M words. The pretrained BERT model can be transferred to various downstream tasks just by joint fine-tuning an added classification layer and all pretrained parameters.

In recent years, the BERT model has also been widely used in the education area and has achieved a significant gain in many application scenarios of intelligent education. For example, Wang *et al.* [40] proposed a hierarchical course BERT model to better capture the course structure quality and linguistic features in each course for predicting teachers' performance in online education. Khodeir [41] combined BERT with a multilayer bidirectional gated recurrent unit to build an urgent classification model for teachers to quickly pick and respond to the most urgent student posts in massive open online course (MOOC) forums, which achieved urgent post classification with a weighted F-score of 91.9%, 91.0%, and 90.0% on three Stanford MOOC Post datasets. Sung *et al.* [42] utilized BERT to build a multilabel classification model for quick assessing the multimodal representational thinking of students in the process to explore the unseen world of thermodynamics. In terms of the ASAG applications, Sung *et al.* [20] analyzed and compared the BERT model with various classical neural network models in short-answer grading, Leon *et al.* [19] analyzed and compared the performance of the BERT model and its variant models, such as AlBERT [43] and robustly optimized BERT pretraining approach (RoBERTa) [31], in short-answer grading.

## III. Methodology

### A. Task Definition

The ASAG problem usually appears in the following two forms [5].

1) *Regression Task:* The students' answer is assigned a score based on the extent of similarity with the corresponding reference answer.

2) *Classification Task:* The students' answer is classified into one category from the set of cognitive categories, i.e., "correct," "partially correct incomplete," "contradictory," "irrelevant," "nondomain" based on the semantic representation of the students' answer and the reference answer.

In this article, we merge the regression task into the classification task by setting grade categories for the scores of students' answers. For example, in the used regression task dataset Mohler [2], for the students' answers whose scores are marked as 0 to 5, we take 11 score grade categories at 0.5 intervals, so that different score values are converted to the corresponding grade categories. For example, a score of 0 corresponds to category 1, the scores in the interval (0, 0.5] correspond to category 2, the scores in the interval (0.5, 1] correspond to category 3, and so on, the scores in the interval (4.5, 5] correspond to category 11.

Accordingly, for any question $T$ and its reference answer $p$, let the grade category set be $Y$. Then, for a given students' answer q to the question $T$, the grading process of the students' answer $q$ is transformed into using a model to predict the probability distribution $\Pr(y|(\mathrm{q},\ \mathrm{p}))$ where $\mathrm{y} \in \mathrm{Y}$, and the final classification label $\mathrm{y}^*$ of the students' answer $q$ is calculated as follows:

$$y^* = \underset{y \in Y}{\operatorname{argmax}}(\Pr(y|(q,p))) \ . \tag{1}$$

### B. Proposed Deep Neural Network Model

Stacked transformers in the BERT can capture very deep semantics, but we argue that for a given input sequence, some classical neural networks, such as Bi-LSTM and Capsule networks, may extract certain fine semantics that cannot be obtained by the transformer, thereby realizing the semantic complementarity and mutual promotion between the classical neural network and the BERT model. First, for the sake of saving resources, transformers in BERT encode the hidden state for a word only using a weighted sum of all other word embeddings, which makes full use of the relations among all words but does not consider the sequence and distance [44]. Therefore, it is necessary to use the memory cells and various gate structures in Bi-LSTM to generate finer global context information to compensate for the lack of temporal information in BERT encoding, and further use the convolutional kernels in a CNN or Capsule network to extract the related local context for each hidden state in BERT. Second, BERT, which is pretrained by large-scale unsupervised learning and fine-tuned by downstream tasks, can provide its upper network with dynamic word embeddings that can change from sentence to sentence. Compared with traditional static word embeddings, such as Glove, the dynamic word embeddings provided by BERT contain richer general-purpose knowledge [45] that can facilitate the training and convergence of upper-layer neural networks. Therefore, the classical neural network on BERT can achieve good performance with less corpus [46], [47]. Third, some studies have shown that in some special tasks with only a few thousand training corpora, such as aspect-level sentiment analysis, further application of classical neural networks on the fine-tuned BERT model can achieve better results. For example, Liao *et al.* [30] combined RoBERTa [31] with a special CNN to improve aspect-category sentiment analysis, Yang *et al.* [32] employed a multihead attention on a fine-tuned BERT model to add distance weights for Chinese-oriented aspect polarity classification, and Nguyen *et al.* [35] stacked additional CNN on the top of BERT for the prediction of extraction information with limited data in domain-specific business documents.

Based on the above-mentioned reasons, this article proposes a novel BERT-based deep neural network model to deeply understand students' answers in the grading task. Specifically, on top of the BERT model, we construct a semantic refinement layer that consists of a Bi-LSTM network and a Capsule network [21] with position information, in which we employ the Bi-LSTM network to extract fine global context for the BERT outputs and employ the Capsule network to extract the related local context for the hidden states of the BERT layer. The framework of our model is shown in Fig. 2, which is described in detail below.

*1) Fined-Tuned BERT Layer:* The BERT layer in our model operates in a fine-tuned manner rather than a dynamic embedding-only approach. All parameters of the BERT layer are initialized from a pretrained BERT model $\mathrm{BERT}_{\mathrm{BASE}}$ [18], but these parameters still require being jointly fine-tuned with other layers in the model.

The input sequence of the fine-tuned BERT is a sentence pair composed of the students' answer and the reference answer as follows:

$$\begin{aligned} \mathrm{S} \ =& \left\{w_{[\mathrm{cls}]}, w_1^q, w_2^q, \ldots, w_u^q, w_{[\mathrm{sep}]},\ w_1^p, w_2^p, \ldots, w_v^p, w_{[\mathrm{sep}]}\right\} \\ & \in \mathbb{R}^{n \times d_w} \end{aligned} \tag{2}$$

where $\{w_i^q\}$, $\{w_i^p\}$, $w_{[\mathrm{cls}]}$, and $w_{[\mathrm{sep}]}$ are the token encodings in BERT, in which $\{w_i^q\}$ corresponds to the tokens in the students' answer, $\{w_i^p\}$ corresponds to the tokens in the reference answer, $w_{[\mathrm{cls}]}$ corresponds to token (CLS) that is the first token of every sequence in BERT, $w_{[\mathrm{sep}]}$ corresponds to token (SEP) that is a delimiter of sentence pairs in BERT. $u$ and $v$ are the numbers of tokens, respectively, in the students' answer and the reference answer. $d_w$ is the dimension of token encoding in BERT, $n$ is the length of the input sequence $s$, and $n = u + v + 3$.

After processing by the BERT layer, we can obtain the preliminary outputs of the input sequence $s$ as follows:

$$O_{\mathrm{BERT}} = \ \mathrm{BERT}(s) \ = \ \left\{h_1^b, h_2^b, \ldots, h_n^b\right\} \in \mathbb{R}^{n \times d_b} \tag{3}$$

where $O_{\mathrm{BERT}}$ denotes the output of the BERT Layer, $\{h_i^b \in \mathbb{R}^{d_b}\}$ denotes the hidden states of the BERT layer, and $d_b$ is the number of hidden units in the BERT layer.

*2) Semantic Refinement Layer:* The semantic refinement layer consists of a standard Bi-LSTM network and a Capsule network [21] with position information. Specifically, we employ the complex gate structures in the Bi-LSTM network to extract fine global context for the BERT outputs and employ convolutional capsules in the Capsule network to extract the

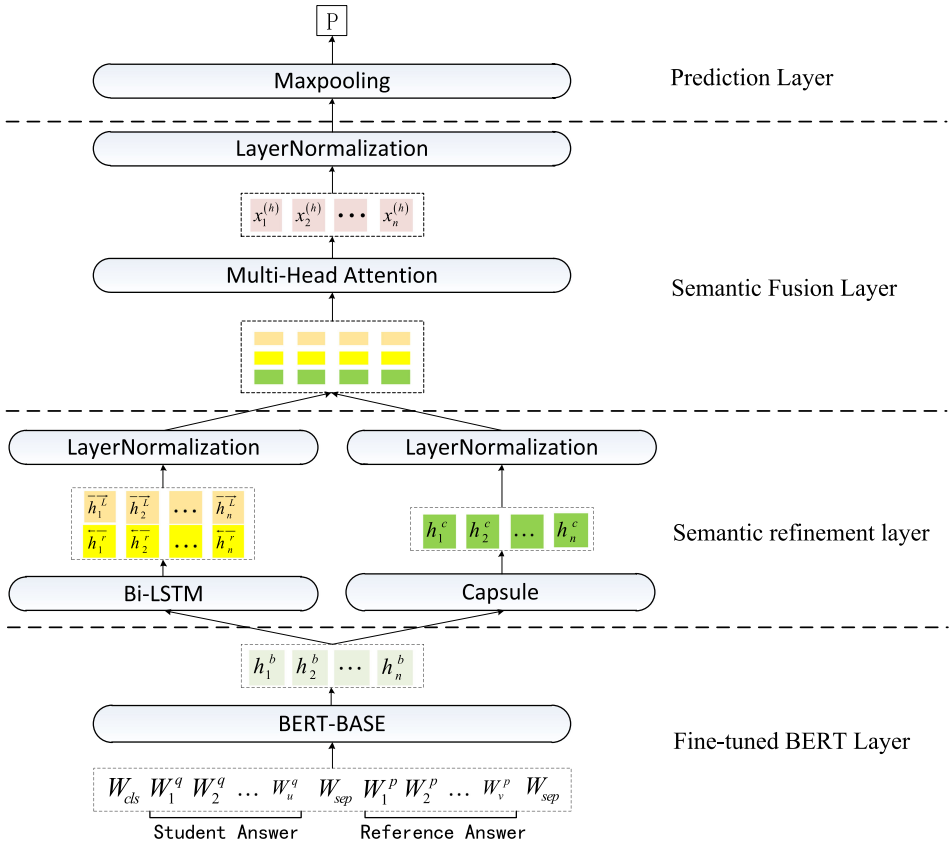


Fig. 2. BERT-based deep neural network model framework for ASAG.

related local context for the hidden states of the BERT layer. Both the Bi-LSTM network and the Capsule network work in parallel, and their outputs are calculated as follows:

$$\overrightarrow{O_{\text{LSTMs}}} = \overrightarrow{\text{LSTMs}}(O_{\text{BERT}}) = \left\{ \overrightarrow{h_1^L}, \overrightarrow{h_2^L}, \ldots, \overrightarrow{h_n^L} \right\} \in \mathbb{R}^{n \times d_L} \quad (4)$$

$$\overleftarrow{O_{\text{LSTMs}}} = \overleftarrow{\text{LSTMs}}(O_{\text{BERT}}) = \left\{ \overleftarrow{h_1^r}, \overleftarrow{h_2^r}, \ldots, \overleftarrow{h_n^r} \right\} \in \mathbb{R}^{n \times d_L} \quad (5)$$

$$O_{\text{Caps}} = \text{Capsules}(O_{\text{BERT}}) = \left\{ h_1^c, h_2^c, \ldots, h_n^c \right\} \in \mathbb{R}^{n \times d_c} \quad (6)$$

where $\overrightarrow{O_{\text{LSTMs}}}$ and $\overleftarrow{O_{\text{LSTMs}}}$ denote the outputs of LSTM networks, respectively, from left to right and from right to left; $O_{\text{Caps}}$ denotes the output of the Capsule network; $\{\overrightarrow{h_i^L}\} \in \mathbb{R}^{d_L}$ and $\{\overleftarrow{h_i^r} \in \mathbb{R}^{d_L}\}$ denote the hidden states of LSTM networks, respectively, from left to right and from right to left; $\{h_i^c \in \mathbb{R}^{d_c}\}$ denotes the output vectors of the convolutional capsule layer in the capsule network; $d_L$ is the number of hidden units in the $\overrightarrow{\text{LSTMs}}$ network and the $\overleftarrow{\text{LSTMs}}$ network; and $d_c$ is the number of convolution cores in the Capsule network.

After the semantic outputs $\overrightarrow{O_{\text{LSTMs}}}$, $\overleftarrow{O_{\text{LSTMs}}}$ and $O_{\text{Caps}}$ are processed by the layer normalization [38], and they are fed into the semantic fusion layer.

*3) Semantic Fusion Layer:* We introduce a semantic fusion layer to integrate the fine token representation and the corresponding local information extracted in the fine semantic layer, in which the outputs of Bi-LSTM and Capsule networks are combined by connecting. First, we stack the three semantic sequences $\overrightarrow{O_{\text{LSTMs}}}$, $\overleftarrow{O_{\text{LSTMs}}}$, and $O_{\text{Caps}}$ into the following matrix:

$$X^{(e)} = \left\{ x_1^{(e)}, x_2^{(e)}, \ldots, x_n^{(e)} \right\} \in \mathbb{R}^{n \times d} \quad (7)$$

where $x_i^{(e)} = [\overrightarrow{h_i^L}; \overleftarrow{h_i^r}; h_i^c] \in \mathbb{R}^d$, $d = 2d_L + d_c$, and [;] is the concatenation operation. We define $[\overrightarrow{h_i^L}; \overleftarrow{h_i^r}]$ as the fine global context for the $i$th token in the input sequence and $h_i^c$ as the related local context for the $i$th token. We connect the $\overrightarrow{h_i^L}$, $\overleftarrow{h_i^r}$, and $h_i^c$ corresponding to the $i$th token in the input sequence end-to-end to form a new hidden representation with a longer dimension, thereby generating a powerful semantic representation for ASAG.

Then, $X^{(e)}$ is fed into a multihead self-attention [25] layer in which each head is a scaled dot-product attention, three inputs of the queries $Q$, keys $K$, and values $V$ are equal to $X^{(e)}$, and the number of heads is equal to $h$. Finally, the fused semantic representation $X^{(h)} \in \mathbb{R}^{n \times d}$ for the answer pair ($q$, $p$) is calculated as follows:

$$\text{MultiHead}(Q, K, V) = [\text{head}_1; \text{head}_2; \ldots; \text{head}_h] w^R \quad (8)$$

$$\text{head}_i = \text{Attention}(Q_i, K_i, V_i) = \text{Attention}\left(Q w^Q, K w^K, V w^V\right) \quad (9)$$

$$\text{Attention}(Q_i, K_i, V_i) = \text{softmat}\left(\frac{Q_i K_i^T}{\sqrt{d_K}}\right) V_i \quad (10)$$

$$X^{(h)} = \text{MultiHead}\left(X^{(e)}, X^{(e)}, X^{(e)}\right) = \left\{ x_1^{(h)}, x_2^{(h)}, \ldots, x_n^{(h)} \right\} \in \mathbb{R}^{n \times d} \quad (11)$$

where (8)–(10) give the calculation process of the multihead attention mechanism. Equation (11) gives the fused result

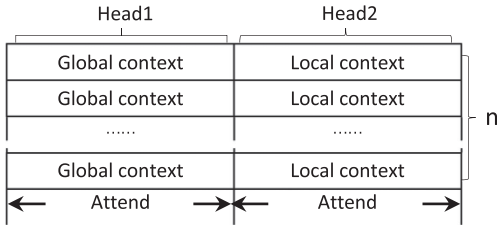


Fig. 3. Schematic diagram of the multihead attention in the semantic fusion layer.

$X^{(h)} \in \mathbb{R}^{n\times d}$obtained by (8)–(10) at the semantic fusion layer, in which $w^Q \in \mathbb{R}^{d\times d_k}$, $w^K \in \mathbb{R}^{d\times d_k}$, $w^V \in \mathbb{R}^{d\times d_V}$, and $w^R \in \mathbb{R}^{d\times d}$ are learnable parameters, $Q = K = V = X^{(e)}$, and $d_K = d_V = d/h$.

In particular, to ensure that the fine global context and the corresponding local information do not interfere with each other, we take the following steps in the multihead attention.

1) We set that the fine global context from the hidden states of Bi-LSTM and the corresponding local context from the Capsule network have equal dimensions, that is, $d_c = 2d_L$.
2) We take the number of heads in the multihead attention to be equal to 2.
3) Each of the fine global context and the corresponding local context is attended by an independent attention in their respective subspaces, and their contributions to the fused semantics are automatically adjusted by the weight matrix $w^R$ in (8), as shown in Fig. 3.

After being processed by the layer normalization, the fused semantic representation $X^{(h)}$ is fed into the prediction layer.

*4) Prediction Layer:* In the prediction layer, we first perform a max-pooling operation on the fused semantic representation $X^{(h)}$ to get the final semantic representation $Z \in \mathbb{R}^d$ for the answer pair (q, p):

$$Z = \text{Maxpooling}\left(X^{(h)}\right) = \{z_1, z_2, \ldots, z_d\} \in \mathbb{R}^d \quad (12)$$

where any $z_j$in $Z$ is calculated as follows:

$$z_j = \text{Max}\left(x_{1j}^{(h)}, x_{2j}^{(h)}, \ldots, x_{nj}^{(h)}\right),\ j = 1, 2, \ldots, d. \quad (13)$$

Then, the final answer pair representation $Z$ is fed into a linear transformation followed by a softmax function to calculate the probability of each grade category:

$$o = MZ + b \quad (14)$$

$$p(y|Z) = \frac{\exp(o_y)}{\sum_i^{d_y} \exp(o_i)} \quad (15)$$

where $M \in \mathbb{R}^{d_y\times d}$ is the representation matrix of grade categories, $b \in \mathbb{R}^{d_y}$ is a bias vector, and $d_y$ is the number of grade categories. $o$ is the vector of confident scores associated with all grade categories, and $p(y\,|\,Z)$ denotes the predicted probability of category $y$ for the given answer pair representation $Z$. To prevent overfitting, dropout [33] is employed in (14).

TABLE II
DETAILS OF TWO DATASETS USED FOR EVALUATION

| | Total | training set |
|---|---|---|
| SemEval-2013 dataset | 10,000 | 4,969 |
| Mohler dataset | 2,273 | 2083*2<br>Randomly select in 12-fold cross validation and double-expand the training set using a correct student answer. |

*5) Loss Function:* To adapt to two different ASAG tasks, we propose two loss strategies for our model. One is the standard cross-entropy loss function using one-hot multinomial distribution for classification tasks in the ASAG problem, which maps the answer pair representation $Z$ to a one-hot gold distribution (. . ., 0, 1, 0, . . .), as shown in (16). The second is our proposed triple-hot loss function for regression tasks, which changes the gold label representation from "one-hot" to "triple-hot": If the label is $y$, the left and right neighbors of $y$ in the distribution vector are also ones, and the others are zeros, i.e., (. . .,0, 1, 1, 1, 0. . .), as shown in (17). The proposed triple-hot loss strategy is based on the following reality: In regression tasks, the true score of the students' answer is usually the average of the scores given by two or more teachers [2], so we argue that the label of the students' answer should correspond to multiple adjacent scoring intervals instead of one.

We define two loss functions for our model as follows:

$$L(\theta) = -\sum_{i=1}^{|\Omega|} \log(p(y_i|Z_i, \theta)) \quad (16)$$

$$L(\theta) = -\sum_{i=1}^{|\Omega|} (\log(p(y_i^{-1}|Z_i, \theta)) + \log(p(y_i|Z_i, \theta)) + \log(p(y_i^{+1}|Z_i, \theta))) \quad (17)$$

where $\theta$ is regarded as all the parameters of our model. $\Omega$ denotes the set of sentence pairs of the students' answer and the reference answer in the training data. $Z_i$ is the representation vector for the $i$th answer pair in $\Omega$. $y_i$ is the grade label of the $i$th answer pair in $\Omega$. $y_i^{-1}$ and $y_i^{+1}$, respectively, represent the left and right adjacent elements of $y_i$ in the sorted grade category list $Y$ for a given regression task.

## IV. EXPERIMENTS

### A. Datasets

We evaluated our model primarily on two widely used datasets, including a SemEval-2013 [17] dataset for the ASAG cognitive classification task and a Mohler dataset [2] for the ASAG regression task, as shown in Table II.

*1) SemEval-2013 Dataset* [17]: We used the SciEntsBank corpus in the SemEval-2013 dataset. SciEntsBank corpus contains approximately 10 000 answers to 197 assessment questions in 15 different science domains. This corpus is a benchmark for the ASAG classification task and is widely evaluated in many works [1], [3], [6], [8], [9], [23]. It involves three classification subtasks on two-way (Correct and Incorrect), three-way (Correct, Contradictory, and Incorrect), and five-

way (Correct, Partially correct, Contradictory, Irrelevant, and Non-domain). To provide multiperspective evaluation, the test dataset is divided into the following three subsets.

1) Unseen Answers (UA): The test data have the same questions and reference answers as the training data and only their students' answers are different.
2) Unseen Questions (UQ): Testing questions and training questions are different, but they still belong to the same domain.
3) Unseen Domains (UD): Testing questions and training questions belong to different domains.

For this dataset, we evaluated our model by reporting accuracy (Acc), weighted-F1 (W-F1), and macro-average F1 (M-F1) on the three-way and five-way subtasks.

*2) Mohler dataset*[1] [2]: A computer science short-answer dataset were created by Mohler *et al.* from ten assignments and two exams of an introductory computer science class at the University of North Texas. It contains 80 questions with 2273 students' answers. Each student's answer was scored by two teachers on an integer ranging from 0 to 5. We took the average of two labeled scores as the true score of the students' answer, resulting in 11 scoring grades ranging from 0 to 5 with 0.5 intervals. The dataset is a benchmark for the ASAG regression task and is widely evaluated in many works [1]–[3], [6], [12], [26], [34]. In our work, we transformed the regression task on the Mohler dataset into a classification task by setting 11 grade categories.

The Mohler dataset contains only 2273 answer pairs, which is too few for deep learning models. Kumar *et al.* [12] proposed an approach to extend the training data of the Mohler dataset for their Bi-LSTM model. They used all the correct students' answers in the training set as the additional reference answers to the questions, thus extending the training pairs from 2100 to about 30 000. To avoid overfitting, we proposed a compromise method, which only selects a correct students' answer from each question as a supplementary reference answer and expands the training set from 2083 pairs to about 3300 pairs (there is no correct students' answer in a few questions). For the Mohler dataset, we evaluated our model in 12-fold cross validation and reported the Cohen's kappa coefficient (kappa), Pearson correlation coefficient (Pearson's $r$), mean absolute error (MAE), and root-mean-square error (RMSE) between the predicted score and the true score.

### B. Experimental Settings

We used $\text{BERT}_{\text{BASE}}$ version[2] (12 layers, 768 units, 12 heads, total parameters = 110M) as the pretraining model of our BERT layer. We set the number of hidden units in each LSTM network to 200 and return all hidden states at the last time step. We set the number of convolution cores in the Capsule network to 400, the size of each convolution core to 3, and the number of dynamic routes in the Capsule network to 3. In the multihead attention, the number of attention heads is set to 2, and each head is assigned 400-dimensional parameters. We set the dropout rate to 0.1 in the Bi-LSTM network, the Capsule network, and the prediction layer. To minimize the loss value, we used the Adam optimizer and set the learning rate to 2e-5. The minibatch size is set to 64. During the training, we increased the epochs from 5 to 10.

[1] [Online]. Available: http://web.eecs.umich.edu/~mihalcea/downloads/ShortAnswerGrading_v2.0.zip

[2] [Online]. Available: https://storage.googleapis.com/bert_models/2020_02_20/uncased_L-12_H-768_A-12.zip

TABLE III
RESULTS OF ABLATION STUDIES

| | SemEval-2013 dataset (Accuracy in 3-way) | | | Mohler dataset (Double-expansion) | | |
|---|---|---|---|---|---|---|
| | UA | UQ | UD | MAE | RMSE | Pearson $r$ |
| Our model | 76.5 | 69.2 | 66.0 | 0.248 | 0.827 | 0.897 |
| w/o refinement | 74.0 | 65.7 | 65.7 | 0.507 | 1.296 | 0.801 |
| w/o Bi-LSTM with BERT | 72.3 | 66.3 | 65.3 | 0.254 | 0.833 | 0.884 |
| w/o Capsule with BERT | 70.1 | 67.2 | 65.1 | 0.349 | 0.928 | 0.869 |
| w/o Capsule with CNN | 70.1 | 68.8 | 63.4 | 0.309 | 0.821 | 0.884 |
| w/o multi-head | 70.9 | 68.4 | 64.5 | 0.483 | 1.341 | 0.751 |
| w/o triple-hot loss | - | - | - | 0.358 | 1.086 | 0.837 |

*Note:* Lower is better for MAE and RMSE; higher is better for accuracy and Pearson's $r$.

### C. Ablation Studies

To analyze the role played by each module in our model, we conducted six aspects of ablation experiments as follows.

1) *W/O Refinement:* It means that the semantic refinement layer is removed from our model and the model degenerates into a fine-tuned BERTBASE.
2) *W/O Multihead:* It means that the multihead self-attention in the semantic fusion layer is removed from our model.
3) *W/O Bi-LSTM With BERT*: It means that the BiLSTM network in the semantic refinement layer is removed from our model and its output is replaced by the output of the BERT layer.
4) *W/O Capsule With BERT:* It means that the Capsule network in the semantic refinement layer is removed from our model and its output is replaced by the output of the BERT layer.
5) *W/O Capsule With CNN:* It means that the Capsule network in the semantic refinement layer is removed from our model and replaced by a CNN network.
6) *W/O Triple-Hot Loss:* It means that the triple-hot loss function for the Mohler dataset is removed from our model and replaced by the standard one-hot loss function.

The results of the ablation study are shown in Table III.

### D. Comparison With Baseline Systems

We compare our model with the following baseline systems for ASAG.

*1) ETS* [8]: One of the better systems reported in SemEval2013-Task7. It uses $n$-gram features and edit features to achieve domain adaptation and stacking for improving short answer scoring.

*2) SOFTCAR* [9]: The best system reported in SemEval2013-Task7. It is based on text overlap through soft cardinality and weight propagation.

TABLE IV
EXPERIMENTS ON THE MOHLER DATASET

| Description | System/Model | MAE | RMSE | Pearson r | Kappa |
|---|---|---|---|---|---|
| Feature engineering systems | Tf-idf (cited in [12]) | - | 1.022 | 0.327 | - |
| | Mohler [2] | - | 0.978 | 0.518 | - |
| | Graph [34] | - | 0.86 | 0.61 | - |
| | ITL [6] | 0.82 | - | - | - |
| Combining sparse features with deep learning | Sultan [3] | - | 0.85 | 0.63 | - |
| | TOKSEN [1] | - | 0.921 | 0.542 | - |
| Word embedding-based deep learning (in this work except for LSTM-EMD) | Bi-LSTM+Capsule | 1.100 | 2.054 | 0.507 | 0.23 |
| | Bi-LSTM+CNN | 1.110 | 2.140 | 0.517 | 0.27 |
| | CNN | 1.769 | 2.866 | 0.002 | 0.02 |
| | CAPSULE | 2.063 | 3.166 | 0.070 | 0.10 |
| | Bi-LSTM | 2.283 | 3.269 | 0.092 | 0.02 |
| | LSTM-EMD$^{++}$ [12] | 0.657 | 1.135 | 0.649 | - |
| Dynamic embedding-only models without Fine-tuning | ELMo (cited in [26]) | - | 0.978 | 0.485 | - |
| | GPT (cited in [26]) | - | 1.082 | 0.248 | - |
| | BERT (cited in [26]) | - | 1.057 | 0.318 | - |
| | GPT-2 (cited in [26]) | - | 1.065 | 0.311 | - |
| Fine-tuned pre-training models (in this work) | Fine-tuned $BERT_{BASE}$ | 0.688 | 1.625 | 0.719 | 0.62 |
| | Fine-tuned $BERT^{+}_{BASE}$ | 0.507 | 1.296 | 0.801 | 0.70 |
| BERT-based deep neural networks | Our model | 0.645 | 1.485 | 0.735 | 0.80 |
| | Our model$^{+}$ | 0.248 | 0.827 | 0.897 | 0.82 |

*Note:* Lower is better for MAE and RMSE; higher is better for Pearson's *r* and Kappa. We compared our model with baseline systems [system name without + means using the original dataset (2273), with a single + means using a double-expanded training set (3333), with ++ means using a fully expanded training set (30 000)].

*3) Mohler* [2]: It learns to grade short-answer questions using knowledge-based similarity measures and dependency graph alignments. It proposed the Mohler dataset and reported the evaluation results in this dataset.

*4) Graph* [34]: It uses graph-based lexico-semantic text matching for short-answer scoring.

*5) Iterative Transfer Learning* (ITL) [6]: It use sparse features to train two classifiers for ASAG. The first one is related to a specific domain. The second classifier is transferable and based on real-valued features capturing from similarities between students' answers and reference answers.

*6) Sultan* [3]: It utilizes augment text similarity features with word embeddings for grading short-answer questions. We directly quoted the results it reported on the Mohler dataset and its results reported by Saha *et al.* [1] on the SemEval-2013 Dataset.

*7) Token Overlap and Sentence Embedding (TOKSEN )* [1]: It uses a pretrained neural network InferSent [22] for encoding the students' answers, which makes up for the lack of contextual representation in the token overlap-based methods.

*8) GCNs* [23]: It combines GCNs with various sparse features for ASAG.

*9) LSTM-EMD* [12]: One of deep learning methods that use a Bi-LSTM framework for ASAG using a pooling layer based on EMD.

*10) Dynamic Embedding-Only Models* [26]: These use various transfer learning models that are pretrained but not fine-tuned, such as ELMo [27], GPT [28], BERT [18], and GPT-2 [29], to extract dynamic sentence embedding for the similarity regression task on the Mohler dataset.

*11) Bi-LSTM+Capsule (CNN)*: To compare with traditional deep learning, we used Glove[3] word embedding instead of the BERT layer in our model to form a deep learning model based on word embedding with the same structure as our model. We also gave experimental results based on individual Bi-LSTM, Capsule, or CNN with the standard cross-entropy loss function.

*12) Fine-Tuned BERTBASE:* To compare with the BERT model, we only added a classification layer on top of the pretrained BERTBASE and jointly fine-tuned the classification layer and all pretrained parameters for ASAG.

Table IV shows the performance of our model on the Mohler dataset with 12-fold training and the comparison between our model and baseline systems. Table V shows the experimental results of our model compared with baseline systems on SemEval-2013 datasets. Figs. 4–6 show the precision–recall (PR) curves of various models on the SemEval-2013 dataset, and Fig. 7 shows the PR curves on the Mohler dataset. Table VI gives the area under the curve (AUC) of the PR curves, and Table VII gives the balance point of the PR curves when precision and recall are equal.

## V. DISCUSSIONS

In Table III, the ablation result of *w/o refinement* shows that our semantic refinement layer significantly improves the grading accuracy of the BERT model on the Mohler dataset and the SemEval-2013 UQ subset. As with the UQ subset in the SemEval-2013 dataset, the testing questions and training questions on the Mohler dataset originate from the same domain. Therefore, this ablation result means that our semantic refinement layer can significantly improve the generalization ability of the BERT model to the features in the domain. The ablation result of *w/o Bi-LSTM with BERT* shows that after directly replacing Bi-LSTM with the output of BERT, the grading accuracy of the model on two datasets has decreased to varying degrees. This means that the complex gate structures in the Bi-LSTM network can extract more refined context information

[3] [Online]. Available: https://github.com/maciejkula/glove-python

TABLE V
COMPARISON RESULTS ON THE SEMEVAL-2013 DATASET

| Description | System /Model | | Three-way | | | | Five-way | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | UA | UQ | UD | Mean | UA | UQ | UD | Mean |
| Feature engineering | ETS | Acc | 72.0 | 58.3 | 54.3 | 61.5 | 64.3 | 43.2 | 44.1 | 50.5 |
| | (cited in [1]) | W-F1 | 70.8 | 53.7 | 46.1 | 56.9 | 64.0 | 41.1 | 41.4 | 48.8 |
| | | M-F1 | 64.7 | 39.3 | 33.3 | 45.8 | 47.8 | 26.3 | 38.0 | 37.4 |
| | SOFTCAR | Acc | 65.9 | 65.2 | 63.7 | 64.9 | 54.4 | 52.5 | 51.2 | 52.7 |
| | (cited in [1]) | W-F1 | 64.7 | 63.4 | 62.0 | 63.4 | 53.7 | 49.2 | 47.1 | 50.0 |
| | | M-F1 | 55.5 | 46.9 | 48.6 | 50.3 | 38.0 | 30.7 | 30.0 | 32.9 |
| | ITL [6] | Acc | - | - | - | - | - | - | - | - |
| | | W-F1 | - | - | - | - | 67.2 | 51.8 | 50.7 | 56.6 |
| | | M-F1 | - | - | - | - | 61.2 | 41.5 | 40.2 | 47.6 |
| Combining sparse features with deep learning | Sultan | Acc | 60.4 | 64.3 | 62.7 | 62.4 | - | - | - | - |
| | (cited in [1]) | W-F1 | 57.0 | 61.5 | 60.3 | 59.6 | - | - | - | - |
| | | M-F1 | 44.4 | 45.5 | 45.2 | 45.0 | - | - | - | - |
| | TOKSEN | Acc | 71.9 | 61.4 | 63.2 | 65.5 | 64.4 | 50.1 | 50.9 | 55.1 |
| | (cited in [1]) | W-F1 | 71.4 | 62.8 | 61.2 | 65.1 | 64.2 | 48.8 | 49.2 | 54.1 |
| | | M-F1 | 66.6 | 49.1 | 47.9 | 54.5 | 48.1 | 31.7 | 35.7 | 38.5 |
| | GCNs [23] | Acc | - | - | 63.4 | - | - | - | 51.2 | - |
| | | W-F1 | - | - | 62.5 | - | - | - | 49.2 | - |
| | | M-F1 | - | - | 56.6 | - | - | - | 46.2 | - |
| Word embedding-based deep learning (in this work) | Bi-LSTM+CNN | Acc | 60.7 | 46.7 | 46.6 | 51.3 | 51.4 | 27.9 | 33.7 | 37.6 |
| | | W-F1 | 58.4 | 44.6 | 39.4 | 47.5 | 49.4 | 23.4 | 29.3 | 34.0 |
| | | M-F1 | 50.3 | 35.6 | 30.1 | 38.7 | 33.6 | 16.9 | 17.9 | 22.8 |
| | Bi-LSTM+ | Acc | 54.2 | 54.6 | 51.2 | 53.3 | 42.3 | 46.2 | 40.0 | 42.8 |
| | Capsule | W-F1 | 51.3 | 52.3 | 46.9 | 50.2 | 33.8 | 34.1 | 32.8 | 33.5 |
| | | M-F1 | 38.5 | 38.9 | 33.7 | 37.0 | 19.5 | 25.6 | 18.1 | 21.1 |
| | Bi-LSTM | Acc | 53.7 | 45.7 | 58.5 | 52.6 | 42.2 | 41.0 | 37.0 | 40.1 |
| | | W-F1 | 49.4 | 42.2 | 55.9 | 49.2 | 37.3 | 34.0 | 33.1 | 34.8 |
| | | M-F1 | 36.7 | 30.7 | 41.3 | 36.2 | 23.5 | 25.0 | 24.0 | 24.2 |
| | Capsule | Acc | 49.8 | 51.5 | 48.9 | 50.1 | 43.7 | 41.7 | 40.4 | 41.9 |
| | | W-F1 | 39.5 | 42.4 | 32.2 | 38.0 | 38.9 | 32.2 | 31.5 | 34.2 |
| | | M-F1 | 29.0 | 29.5 | 22.0 | 26.8 | 24.3 | 22.5 | 17.3 | 21.4 |
| Dynamic embedding-only (in this work) | $BERT_{BASE}$ without fine-tuning | Acc | 52.1 | 47.2 | 54.1 | 51.1 | 45.5 | 44.2 | 44.6 | 44.8 |
| | | W-F1 | 49.5 | 44.3 | 51.2 | 48.3 | 32.8 | 30.2 | 31.5 | 31.5 |
| | | M-F1 | 36.4 | 32.8 | 38.6 | 35.9 | 17.5 | 20.6 | 16.8 | 18.3 |
| Fine-tuned BERT (in this work) | $BERT_{BASE}$ with fine-tuning | Acc | 74.0 | 65.7 | 65.7 | 68.5 | 68.1 | 52.5 | 53.1 | 57.9 |
| | | W-F1 | 72.5 | 64.6 | 64.3 | 67.1 | 64.6 | 51.3 | 53.0 | 56.3 |
| | | M-F1 | 68.5 | 54.5 | 55.7 | 59.6 | 48.5 | 43.1 | 41.0 | 44.2 |
| BERT-based deep neural networks | Our model | Acc | 76.5 | 69.2 | 66.0 | 70.5 | 69.8 | 58.0 | 53.2 | 60.2 |
| | | W-F1 | 75.8 | 67.4 | 64.6 | 69.2 | 69.8 | 55.6 | 54.1 | 59.8 |
| | | M-F1 | 71.4 | 57.9 | 55.9 | 61.7 | 53.2 | 44.2 | 50.4 | 49.2 |

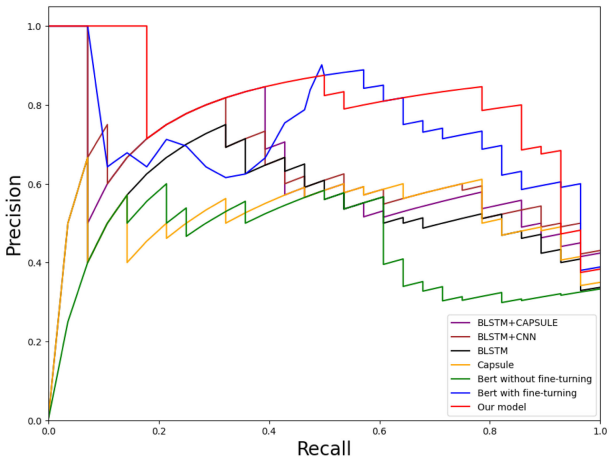


Fig. 4. PR curve of three-way UA on the SemEval-2013 dataset.

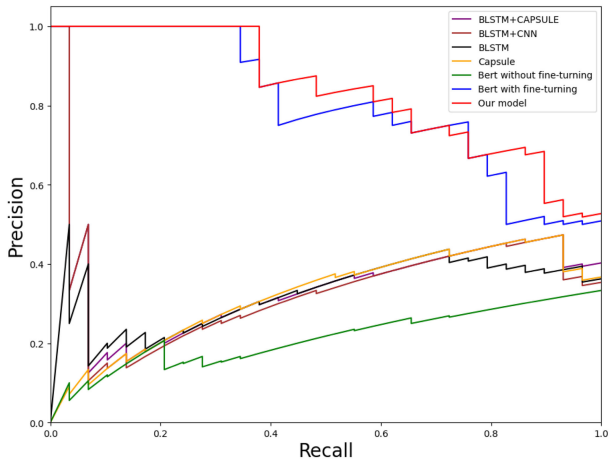


Fig. 5. PR curve of three-way UQ on the SemEval-2013 dataset.

from the output of the BERT model, just as it performs in other natural language processing tasks, such as Named Entity Recognition [48]. The ablation result of *w/o Capsule with BERT* shows that after directly replacing the Capsule network with the output of BERT, the grading accuracy of the model on two datasets has a significant drop. This means that the Capsule network can extract more refined local information from the output of the BERT model. The ablation result of *w/o Capsule*

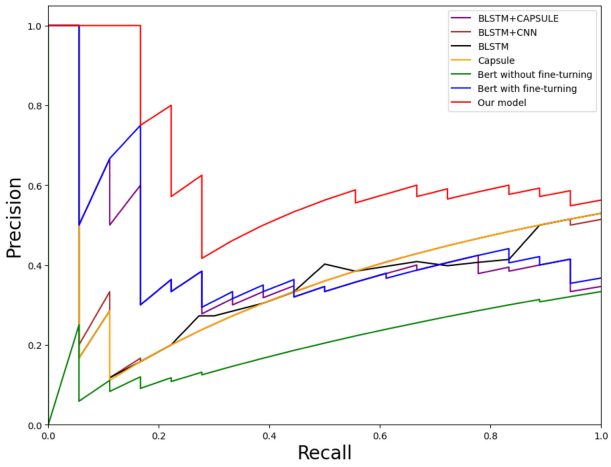


Fig. 6. PR curve of three-way UD on the SemEval-2013 dataset.

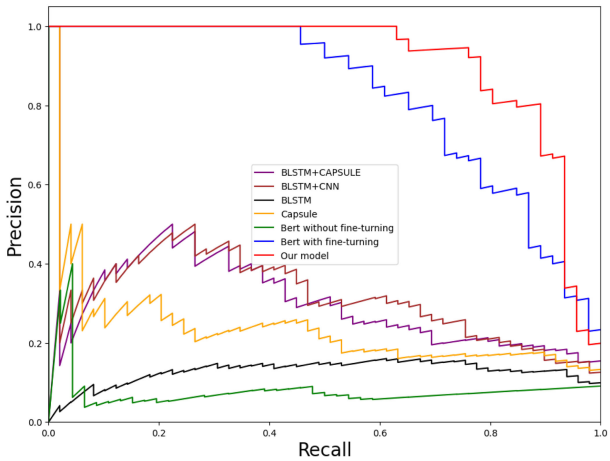


Fig. 7. PR curve on the Mohler dataset.

*with CNN* shows that the performance of the Capsule network with position information is better than that of the ordinary CNN network in the BERT-based deep neural networks. The ablation result of *w/o multi-head* shows that the multihead self-attention in the semantic fusion layer plays an important role in improving the performance of the model. In addition, the ablation result of *w/o triple-hot loss* shows that the proposed triple-hot loss strategy significantly improves the Pearson's $r$ of our model on the Mohler dataset from 0.837 to 0.897.

Table IV gives results comparing our model with various baseline systems on the Mohler dataset, which is a typical benchmark for the ASAG regression task. The experimental results show that the feature engineering-based methods achieve a moderate scoring accuracy in the Mohler dataset, in which the Graph system [34] reported the better Pearson's $r$ and RMSE results of 0.61 and 0.86, respectively. Combining sparse features with deep learning in the Sultan system [3] slightly improves the Pearson's $r$ of feature engineering systems from 0.61 to 0.63. Limited by the lack of training data, the word-embedding-based single neural network classifiers perform very poorly on the original Mohler dataset, and the performances of LSTM-EMD [12] on the fully expanded dataset and Bi-LSTM combined by Capsule or CNN on the original Mohler dataset are only

TABLE VI
AUC OF THE PR CURVE

| System /Model | SemEval-2013 dataset ( in 3-way) | | | Mohler dataset |
|---|---|---|---|---|
| | UA | UQ | UD | |
| Bi-LSTM+Capsule | 0.636 | 0.362 | 0.419 | 0.290 |
| Bi-LSTM+CNN | 0.646 | 0.352 | 0.393 | 0.324 |
| Bi-LSTM | 0.545 | 0.326 | 0.384 | 0.127 |
| Capsule | 0.515 | 0.320 | 0.391 | 0.234 |
| BERT without fine-tuning | 0.430 | 0.213 | 0.204 | 0.078 |
| BERT with fine-tuning | 0.724 | 0.802 | 0.434 | 0.823 |
| Our model | 0.812 | 0.835 | 0.642 | 0.908 |

comparable to that of feature engineering systems on the original dataset. Similarly, dynamic embedding-only models without fine-tuning, such as ELMo, GPT, BERT, and GPT-2, perform very poorly in the similarity regression on the original Mohler dataset. Surprisingly, the fine-tuned BERT model performs very well in the scoring grade classification for the Mohler dataset, and it achieves a Pearson's $r$ of 0.719 on the original dataset or 0.801 on the double-expanded dataset, which reveals that fine-tuning is very important for improving the performance of the BERT model on the Mohler dataset. Finally, the experimental results show that our model, a BERT-based deep neural network, achieves the best Kappa, Pearson's $r$, RMSE, and MAE results of 0.82, 0.897, 0.827, and 0.248, respectively, which significantly surpass the results of all baseline systems on the extended Mohler dataset. Meanwhile, our model also achieves the best performance in the comparison under the original Mohler dataset. By adding a semantic refinement layer, our model significantly improves the Pearson's $r$ and Kappa of the fine-tuned BERT model on the double-expanded dataset from 0.801, 0.70 to 0.897, 0.82, which demonstrates that our semantic refinement layer can significantly improve the ability of the BERT model to generalize features in the domain.

Table V gives results comparing our model with various baseline systems on the SemEval-2013 dataset, which is a typical benchmark for the ASAG classification task. The experimental results show that feature engineering systems have achieved better scoring results on the SemEval-2013 dataset. Some systems report very good results on certain subsets. For example, ETS reports a good scoring accuracy of 72 on the UA subset of three-way, SOFTCAR reported good scoring accuracies on both the UQ and UD subsets, and ITL reported good F1 values on the UA subset of five-way. The overall performance of combining sparse features with deep learning is comparable to that of feature engineering systems. Limited by the lack of training data, the overall performance of word-embedding-based deep learning on the SemEval-2013 dataset is far worse than that of feature engineering systems, which means that self-training neural networks, such as CNN, LSTM, and Capsule cannot play an important role in ASAG classification tasks with small datasets. The experimental results also show that the dynamic-embedding-only BERT model without fine-tuning cannot achieve good results, whereas fine-tuning in the ASAG task can significantly improve BERT's performance, which exceeds the performance of feature engineering systems. Finally, the experimental results show that our model, a BERT-

TABLE VII
BALANCE POINT OF THE PR CURVE

| System /Model | SemEval-2013 dataset (in three-way) | | | Mohler dataset |
|---|---|---|---|---|
| | UA | UQ | UD | |
| Bi-LSTM+Capsule | 0.536 | 0.207 | 0.278 | 0.388 |
| Bi-LSTM+CNN | 0.571 | 0.138 | 0.167 | 0.388 |
| Bi-LSTM | 0.535 | 0.206 | 0.272 | 0.082 |
| Capsule | 0.571 | 0.206 | 0.238 | 0.244 |
| BERT without fine-tuning | 0.536 | 0.206 | 0.111 | 0.065 |
| BERT with fine-tuning | 0.714 | 0.724 | 0.333 | 0.717 |
| Our model | 0.786 | 0.758 | 0.555 | 0.804 |

based deep neural network, outperforms all baseline systems in terms of scoring accuracy. Our model not only significantly exceeds the scoring accuracy of feature engineering systems on each subset but also significantly improves the scoring accuracy of the fine-tuned BERT model on the UQ subset from 65.7 (three-way) and 52.5 (five-way) to 69.2 (three-way) and 58.0 (five-way), which once again demonstrates that our semantic refinement layer can promote the BERT model to deeply understand the semantics of answer texts and improve the ability of the BERT model to generalize features in the domain.

Figs. 4–6 show that the PR curve of our model is much higher than the PR curves of the other models in most cases, which is consistent with the conclusion given in Table VI that our model achieves the largest AUC in all graphs. Meanwhile, Table VII shows that our model reaches the balance point of equal precision and recall with the highest value in all graphs, which means that our model has the best performance among all the compared models. On the other hand, Tables VI and VII also show that the improvement of our model over the fine-tuned BERT model is significant. For example, our model improves the AUC and balance point values of the BERT model by more than 40% on the SemEval-2013 UD dataset and by more than 10% on the SemEval-2013 UA and Mohler datasets.

Our proposed model can be applied to two scenarios in intelligent education systems. First, our model can be applied to the evaluation module in ITSs, which can improve the evaluation of test questions, such as fill-in-the-blank and explanation from manual grading [37] to machine, and realize the automatic association of answer texts with cognitive types. The SemEval-2013 benchmark [17] uses a training set of size 4969 to construct a simulated environment for the evaluation modules of ITSs, in which the UA and UQ subsets simulate the test evaluation for restricted-domain ITSs and the UD subset simulates the test evaluation for open-domain ITSs. The results presented in Table V show that our model performs better on test evaluations for restricted-domain ITSs and worse for open-domain ITSs, which means that our model requires more training corpus in open-domain ITSs. Second, the proposed model can also be applied to MOOCs and online exam platforms. In this application scenario, our model solves the challenge of replacing teachers to quickly and accurately grade a large number of free-text answers provided by the students. The Mohler benchmark [2] constructs a simulated evaluation environment for this application scenario. The results presented in Table IV show that the scoring result of our model achieves an excellent Pearson correlation of 0.897 with human judgment under a training set of size 3333, which means that our model has reached a practical level.

## VI. CONCLUSION

In this article, we proposed a novel BERT-based deep neural network model for ASAG. Through extensive experimental comparison, this article reveals the following theoretical implications:

1) Word-embedding-based self-training neural networks, such as CNN, LSTM, and Capsule, cannot play an important role in ASAG tasks with small datasets.
2) The BERT model, which has a deep transformer coding structure and is pretrained on a large-scale corpus, can perform better by simply fine-tuning in the ASAG task.
3) On the fine-tuned BERT model, the LSTM and Capsule networks can extract more fine global and local semantics to deepen the BERT model's understanding of the answer text and further improve the feature generalization ability of the BERT model in the domain.

Our model still has the following two limitations: 1) In terms of the cognitive evaluation of open-domain ITSs, such as the UD subset in the SemEval-2013 benchmark [17], our model fails to achieve good performance under small corpus training, and it requires more corpus training; and 2) our model is currently unable to eliminate and replace a large number of pronouns present in students' answers. In the next step, we plan to use the BERT model to eliminate pronouns in students' answers to further improve the scoring accuracy of the ASAG task.

## REFERENCES

[1] 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.
[2] M. Mohler, R. Bunescu, and R. Mihalcea, "Learning to grade short answer questions using semantic similarity measures and dependency graph alignments," in *Proc. Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2011, pp. 752–762.
[3] M. A. Sultan, C. Salazar, and T. Sumner, "Fast and easy short answer grading with high accuracy," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2016, pp. 1070–1075.
[4] S. Marvaniya, S. Saha, T. I. Dhamecha, P. Flotz, 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.*, 2018, pp. 993–1002.
[5] 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.–Mar. 2020.
[6] S. Roy, H. S. Bhatt, and Y. Narahari, "An iterative transfer learning based ensemble technique for automatic short answer grading," 2016, *arXiv:1609.04909v3*.
[7] N. Süzen, A. N. Gorban, J. Levesley, and E. M. Mirkes, "Automatic short answer grading and feedback using text mining methods," *Procedia Comput. Sci.*, vol. 169, no. 2020, pp. 726–743, 2020.
[8] M. Heilman and N. Madnani, "ETS: Domain adaptation and stacking for short answer scoring," in *Proc. Joint Conf. Lexical Comput. Semantics*, 2013, vol. 2, pp. 275–279.
[9] S. Jimenez, C. Becerra, and A. Gelbukh, "Softcardinality: Hierarchical text overlap for student response analysis," in *Proc. Joint Conf. Lexical Comput. Semantics,* 2013, vol. 2, pp. 280–284.
[10] G. Jorge-Botana, J. M. Luzón, I. Gómez-Veiga, and M. 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, 2015.

[11] D. Alikaniotis, H. Yannakoudakis, and M. Rei, "Automatic text scoring using neural networks," in *Proc. 54th Annu. Meeting Assoc. Comput Linguistics*, 2016, vol. 1, pp. 715–725.
[12] S. Kumar, S. Chakrabarti, and S. Roy, "Earth movers distance pooling over Siamese LSTMs for automatic short answer grading," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2046–2052.
[13] 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.
[14] 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.
[15] B. Riordan, A. Horbach, A. Cahill, T. Zesch, and M. L. Chong, "Investigating neural architectures for short answer scoring," in *Proc. 12th Workshop Innov. Use NLP Building Educ. Appl.*, 2017, pp. 159–168.
[16] 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.
[17] M. O. Dzikovska *et al.*, "SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge," in *Proc. 2nd Joint Conf. Lexical Comput. Semantics*, 2013, vol. 2, pp. 263–274.
[18] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 4171–4186.
[19] C. Leon and F. Anna, "Investigating transformers for automatic short answer grading," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2020, pp. 43–48.
[20] C. Sung, T. I. Dhamecha, and N. Mukhi, "Improving short answer grading using transformer-based pre-training," in *Proc. Conf. Artif. Intell. Educ.*, 2019, pp. 469–481.
[21] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, and Z. Zhao, "Investigating capsule networks with dynamic routing for text classification," in *Proc. Conf. Empirial Methods Natural Lang. Process.*, 2018, pp. 43–48.
[22] 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.
[23] H. Tan, C. Wang, Q. Duan, Y. Lu, and R. Li, "Automatic short answer grading by encoding student responses via a graph convolutional network," *Interact. Learn. Environ.*, vol. 2020, pp. 1–15, 2020.
[24] 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, 2020.
[25] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[26] S. K. Gaddipati, D. Nair, and P. G. Plger, "Comparative evaluation of pretrained transfer learning models on automatic short answer grading," 2020, *arXiv*:2009.01303v1.
[27] M. E. Peters *et al.*, "Deep contextualized word representations," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 2227–2237.
[28] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding with unsupervised learning," *OpenAI Blog*, 2018. [Online]. Available: https://openai.com/blog/language-unsupervised/
[29] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Better language models and their implications," *OpenAI Blog*, 2019. [Online]. Available: https://openai.com/blog/better-language-models/
[30] W. X. Liao, B. Zeng, X. W. Yin, and P. F. Wei, "An improved aspect-category sentiment analysis model for text sentiment analysis based on RoBERTa," *Appl. Intell.*, vol. 51, pp. 3522–3533, 2021.
[31] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692v1*.
[32] H. Yang, B. Zeng, J. Yang, Y. Song, and R. Xu, "A multi-task learning model for Chinese-oriented aspect polarity classification and aspectterm extraction," *Neurocomputing*, vol. 419, pp. 344–356, 2020.
[33] 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. 56, pp. 1929–1958, 2014.
[34] L. Ramachandran, J. Cheng, and P. Foltz, "Identifying patterns for short answer scoring using graph-based lexico-semantic text matching," *Proc. 10th Workshop Innov. Use NLP Building Educ. Appl.*, 2015, pp. 97–106.
[35] M. T. Nguyen, D. T. Le, and L. Le, "Transformers-based information extraction with limited data for domain-specific business documents," *Eng. Appl. Artif. Intell.*, vol. 97, 2021, Art. no. 104100.
[36] 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, 2020.
[37] S. Drissi and A. Amirat, "An adaptive e-learning system based on student's learning styles: An empirical study," *Int. J. Distance Educ. Technol.*, vol. 14, no. 3, pp. 34–51, 2016.
[38] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[39] I. Katsaris and N. Vidakis, "Adaptive e-learning systems through learning styles: A review of the literature," *Adv. Mobile Learn. Educ. Res.*, vol. 1, no. 2, pp. 124–145, 2021.
[40] W. Wang, H. Zhuang, M. Zhou, H. Liu, and B. Li, "What makes a star teacher? A hierarchical BERT model for evaluating teacher's performance in online education," 2020, *arXiv:2012.01633v1*.
[41] N. Khodeir, "Bi-GRU urgent classification for MOOC discussion forums based on BERT," *IEEE Access*, vol. 9, no. 1, pp. 58243–58255, Apr. 2021.
[42] S. H. Sung, C. Li, G. Chen, X. Huang, and J. Shen, "How does augmented observation facilitate multimodal representational thinking? Applying deep learning to decode complex student construct," *J. Sci. Educ. Technol.*, vol. 30, no. 1, pp. 210–226, 2021.
[43] Z. Lan *et al.*, "Albert: A lite bert for self-supervised learning of language representations," 2019, *arXiv:1909.11942*.
[44] B. Yang, L. Wang, D. F. Wong, L. S. Chao, and Z. Tu, "Convolutional self-attention networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, vol. 1, pp. 4040–4045.
[45] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019, *arXiv:1910.10683v3*.
[46] J. Zhou, X. Huang, Q. Hu, and L. He, "SK-GCN: Modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification," *Knowl.-Based Syst.*, vol. 205, no. 3, 2020, Art. no. 106292.
[47] W. Song, Z. Wen, Z. Xiao, and S. Park, "Semantics perception and refinement network for aspect-based sentiment analysis," *Knowl.-Based Syst.*, vol. 214, 2021, Art. no. 106755.
[48] F. Li, Z. Wang, S. C. Hui, L. Liao, and M. Jia, "Modularized interaction network for named entity recognition," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, 2021, vol. 1, pp. 200–209.

**Xinhua Zhu** received the bachelor's degree in computer science from the Department of Radio Electronics, Beijing Normal University, Beijing, China, in 1989.

He is currently a Professor with the School of Computer Science and Engineering, Guangxi Normal University, Guilin, China. He is also the Principal Investigator for several National Natural Science Foundation Projects. His research interests include intelligent tutoring systems, natural language processing, sentiment analysis, knowledge graphs, and neural computing.

**Han Wu** received the bachelor's degree in Internet of Things engineering from Yancheng Normal University, Yancheng, China, in 1989. He is currently working toward the master's degree in software engineering from the School of Computer Science and Engineering, Guangxi Normal University, Guilin, China.

His research interests include intelligent tutoring systems and neural computing.

**Lanfang Zhang** received the bachelor's and master's degrees from Guangxi Normal University, Guilin, China, in 1989 and 2005, respectively, both in computer engineering.

She is currently a Professor with the Faculty of Education, Guangxi Normal University. She is also the Principal Investigator for the National Natural Science Foundation Project. Her research interests include artificial intelligence in education, natural language processing knowledge graphs, and neural computing.