Grading Chinese Answers on Specialty Subjective Questions

Dongjin Li[1,2], Tianyuan Liu[1,2], Wei Pan[1], Xiaoyue Liu[1], Yuqing Sun[1(✉)], and Feng Yuan[3]

[1] School of Software, Shandong University, Jinan, China
lidongjinl994@163, zodiacg@foxmail.com, panwei_sdu@163, suiqiyue@163.com, sun_yuqing@sdu.edu.cn
[2] School of Computer Science and Technology, Shandong University, Jinan, China
[3] Shandong University Ouma Software Co., Ltd., Jinan, China
sdyuanf@sina.com

Abstract. It is an important task to grade answers on specialty subjective questions, which is helpful for the supervision of human review and improving the efficiency and quality of review process. Since this grading process should be performed at the same time with human review, there are only a few samples available for each question that can be provided by specialty experts before review process. We investigate the problem of grading Chinese answers on specialty subjective questions with a reference answer in this paper by proposing a grading model that combines two Bi-LSTM networks with attention mechanism. The first part is a sequence to sequence Bi-LSTM network that adopts the pre-trained word embeddings as input. Since there is no embedding for some specialty words, we instead use the fine-grained word embeddings. After the max-pooling on each sentence, we adopt the mutual attention mechanism to learn the matching degree on specialty knowledge between each pair of sentences of answer and reference. Then we adopt another Bi-LSTM with max-pooling to have an overall vector. By concatenating these two vectors from answer and reference, a multilayer perceptron is adopted to predicate the scores. We adopt the real datasets on a national specialty examination to thoroughly verify the model performance against different amount of training data, network structures, pooling strategies and attention mechanisms. The experimental results show the effectiveness of our method.

Keywords: Grading Chinese answer · Specialty subjective questions · Attention mechanism

1 Introduction

It is an important task to grade answers on specialty subjective questions. We investigate the problem of grading Chinese answers on specialty subjective questions with a reference answer in this paper. Although there are quite a few works on grading English essays, they are not applicable for our problem due to the following challenges.

Y. Sun et al. (Eds.): ChineseCSCW 2019, CCIS 1042, pp. 670–682, 2019.
https://doi.org/10.1007/978-981-15-1377-0_52

One is the reference answer. In English essay scoring, there is not any reference. For example, the E-rater system developed by Burstein [1] scored English essays from the perspectives of syntactic analysis, subject analysis and other semantic aspects. Instead, in the subjective question problem, we are given the standard answer as reference for each question. When evaluating the student answers, we need to exam how much they match on knowledge points. For a specificity subjective question, the content precisely defines the direction and scope of answer. Some answers hit the key words in reference and seem similar in phrase level, but they might be logically wrong. Many student answers contain the same specialty words such that the evaluation on lexical or even syntax feature does not work.

The second is the insufficient amount of training data. Text classification methods based on deep learning generally require a large number of training samples. In our scoring scene, the model needs to learn based on a small number of labeled samples. Since the exam questions change every year, the data of previous years are not suitable for the grading task of this year.

The third is the discrete scores. Generally, experts examine how many knowledge points are targeted by a student answer, and assign different discrete scores. It is not suitable to directly adopt the classification or regression methods for this scoring process.

There is also another challenge on the specialty word embeddings. The pre-trained universal word embeddings do not exactly contain all specialty words. Since there are not enough specialty corpus, it is difficult to learn stable embeddings for specialty words.

To tackle these challenges, we propose a grading model based on mutual attention mechanism. When a specialty word has no embedding, we use its fine-grained words embeddings to represent the word. We combine bidirectional Long Short-Term Memory (Bi-LSTM) network and mutual attention mechanism to grade student answers, taking into account the semantic information of student answer and its matching degree to the reference answer. We adopt the real datasets on a national specialty examination to thoroughly examine the performance against different amount of training data, network structures, pooling strategies and attention mechanisms. The experimental results show the effectiveness of our proposed method.

The rest of this paper is organized as follows. Section 2 introduces related works. Section 3 introduces our grading model. In Sect. 4, we validate our model on real datasets and analyze the experimental results. Section 5 summaries this paper and presents future work.

2 Related Work

So far, to the best of our knowledge, there is not any publicly available works on the task of grading Chinese answer of specialty questions that are exactly related with our work. In this section, we present some works that are technically related. At present, the Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) are often adopted to extract semantic features from text. Sutskever et al. [2] performed the language translation task using a RNN model based on Long Short-Term Memory

(LSTM) to obtain sentence vector. Colleber et al. [3] and Kim et al. [4] extracted features using CNN, and achieved good results in tasks such as part-of-speech tagging, sentiment classification, and named entity recognition. Zhang et al. [5] used CNN to model sentences at the character level and applied the obtained sentence vector to text classification task. Kalchbrenner et al. [6] proposed the Dynamic Convolutional Neural Network which used dynamic k-max pooling, and the model achieved good performance on multi-class sentiment prediction tasks. Schwenk et al. [7] and Johnson et al. [8] extracted deeper semantic features through multi-level convolution and performed well on text classification.

The combination of CNN and RNN are also adopted to extract the semantic features. Tang et al. [9] generated a sentence vector by extracting features through CNN at lexical level, and then generated a text vector by extracting sentences sequence features based on a Gated Recurrent Unit (GRU) network. Lai et al. [10] used RNN to encode a sentence, and then obtained the sentence vector via pooling operations. Shi et al. [11] replaced the convolution kernel of CNN with LSTM to encode sentence and used the generated vector for text classification. Xiao et al. [12] used CNN and RNN to process sentences respectively, and then concatenated the generated vectors as the sentence vector which was applied to text classification. By using CNN and RNN together, the local features and context-sensitive features of the text are extracted separately.

In the subjective review task with reference, the final score of a student answer is not only determined by features of answer text, but also by the matching degree between the answer and reference. The introduction of attention mechanism has enabled the model to capture the points of focus on each answer. Bahdanau et al. [13] first introduced attention mechanism into natural language processing field in 2014. In machine translation, the authors calculated the related information between current word and each word of the sentence to be translated, and dynamically searched the information related to current word during decoding. The attention mechanism can dynamically acquire the key information focused by current word or sentence, and was later applied to multiple natural language processing tasks such as question answering, text entailment and text classification. In question answering task, Tan et al. [14] generated a representation for a specific question by calculating attention weights of the candidate answers and the question. In the reading comprehension task, Chaturvedi et al. [15] concatenated the question with each candidate answer, and calculated attention weights on each sentence in the context. Yang et al. [16] introduced the attention mechanism into the GRU network on text classification tasks. In the Chinese cloze-style reading comprehension task, Cui et al. [17] proposed a consensus attention mechanism to calculate the attention weights between each words in the query and the document. In the English cloze-style reading comprehension task, Cui et al. [18] proposed a mutual attention mechanism by calculating the text-based attention and the question-based attention respectively, and combing two attention weights as the probability of each word in the text to be the standard answer. We adopt this idea into our model to calculate the matching degree of answer and reference.

3 The Grading Model on Specialty Subjective Questions

3.1 The Grading Model on Specialty Subjective Questions

The grading task on specialty subjective questions with reference answer is defined as follows. For the subjective question Q, the student answer text X_0 and the reference answer text A_0, the problem is to predict student answer's score $c \in C$, where $C = \{c_1, c_2, \ldots, c_r\}$ is a set of categories according to the score range of Q. We propose a grading model based on Bi-LSTM and mutual attention mechanism, which is shown in Fig. 1. Details of our model are given below.

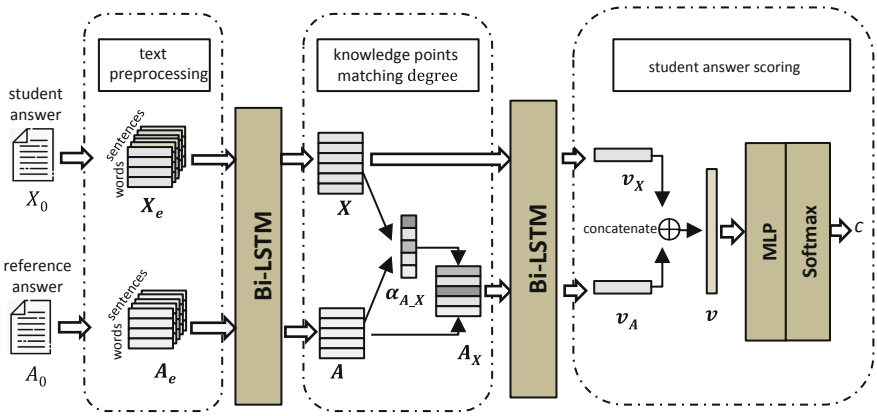

Fig. 1. The grading model on specialty subjective questions.

Text Preprocessing. First, the student answer X_0 and the reference answer A_0 are segmented into sentences according to commas, periods, semicolons and colons. Then each sentence is segmented into words. We adopt pre-trained Chinese word embeddings [19] as the word embedding. Since this is a specialty exam, the text may contain some specialty words which have no embedding. In order to retain the semantic information of the word, we combine the fine-grained word vectors to obtain the embedding of the specialty word. The specialty word without embedding is segmented into subwords. If a subword still has no embedding, the segmentation is performed to the subword again until it has embedding or is split to characters.

After the text preprocessing, we have the embedding form of student answer $\boldsymbol{X_e}$ and the reference answer $\boldsymbol{A_e}$, where $\boldsymbol{X_e} \in R^{m \times l \times d_0}$, $\boldsymbol{A_e} \in R^{n \times l \times d_0}$, m is the number of answer sentences, n is the number of reference sentences, l is the number of words segmented by each sentence after padding, and d_0 is the dimension of word embedding.

The Bi-LSTM Network for Semantic Feature Extraction. We adopt a sequence to sequence Bi-LSTM model to extract the semantic features of both student answer X_e and reference answer A_e with the max-pooling on each sentence. For sentence $\mathbf{s} = [\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_l}]$, where $\boldsymbol{w_l}$ is word embedding of l th word in $\boldsymbol{s}$. The forward LSTM

encodes words sequence along the direction of from the first word to the last word, and the backward LSTM encodes words sequence along the reverse direction.

We adopt the max-pooling on the hidden state vectors of all timesteps of forward LSTM and backward LSTM, respectively, to obtain the forward vector $\vec{\boldsymbol{h}}$ and backward vector $\overleftarrow{\boldsymbol{h}}$. $\vec{\boldsymbol{h}}$ and $\overleftarrow{\boldsymbol{h}}$ are concatenated as the final sentence vector $\boldsymbol{h}$.

$$\vec{\boldsymbol{h}} = \mathrm{LSTM}_f(\boldsymbol{s}),\ \overleftarrow{\boldsymbol{h}} = \mathrm{LSTM}_b(\boldsymbol{s}),\ \boldsymbol{h} = \vec{\boldsymbol{h}} \oplus \overleftarrow{\boldsymbol{h}} \tag{1}$$

For each pair of $\boldsymbol{X_e}$ and $\boldsymbol{A_e}$, we can get the encoded student answer $\boldsymbol{X} = \left[\boldsymbol{h_1^X}, \boldsymbol{h_2^X}, \ldots, \boldsymbol{h_m^X}\right]$ and reference answer $\boldsymbol{A} = \left[\boldsymbol{h_1^A}, \boldsymbol{h_2^A}, \ldots, \boldsymbol{h_n^A}\right]$ by Bi-LSTM, respectively.

$$\boldsymbol{X} = \mathrm{BiLSTM}(\boldsymbol{X_e}),\ \boldsymbol{A} = \mathrm{BiLSTM}(\boldsymbol{A_e}) \tag{2}$$

where $\boldsymbol{X} \in R^{m \times 2d_1}$, $\boldsymbol{A} \in R^{n \times 2d_1}$, d_1 is the number of Bi-LSTM hidden units.

Matching Degree on Knowledge Points Based on Mutual Attention Mechanism. The detection of matching degree on knowledge points between student answer and reference is performed by mutual attention mechanism, which consists of two parts.

The first part is the one-way attention of student answer $\boldsymbol{X}$ to reference answer $\boldsymbol{A}$. For sentence vector $\boldsymbol{h_i^A}$ of i th sentence of $\boldsymbol{A}$, and sentence vector $\boldsymbol{h_j^X}$ of j th sentence of $\boldsymbol{X}$, the matching score $M_{i,j}$ is calculated as follows.

$$M_{i,j} = \boldsymbol{h_i^A} \cdot \boldsymbol{h_j^{X^T}} \tag{3}$$

The matching score for the whole student answer $\boldsymbol{X}$ and reference $\boldsymbol{A}$ is calculated pairwise as the matching score matrix $\boldsymbol{M} \in R^{n \times m}$. The column-wise softmax function is applied to $\boldsymbol{M}$ to obtain the one-way attention matrix $\boldsymbol{\alpha} \in R^{n \times \mathrm{m}}$ of the student answer to reference answer. For sentence $\boldsymbol{h_p^X}$ in student answer $\boldsymbol{X}$, let $\boldsymbol{\alpha}(p)$ represent the distributions of matching degree between $\boldsymbol{h_p^X}$ and each sentence of reference answer $\boldsymbol{A}$.

$$\begin{aligned} \boldsymbol{\alpha}(p) &= softmax\left(M_{1,p}, \ldots, M_{n,p}\right) \\ \boldsymbol{\alpha} &= [\boldsymbol{\alpha}(1), \ldots, \boldsymbol{\alpha}(m)] \end{aligned} \tag{4}$$

The second part is the mutual attention of student answer and reference answer. In the general attention mechanism, each row of the one-way attention matrix $\boldsymbol{\alpha}$ is simply added or averaged as the final attention weights. In this grading task, for the pth sentence $\boldsymbol{h_p^X}$ of student answer $\boldsymbol{X}$, even if the content of the sentence is completely irrelevant to reference answer $\boldsymbol{A}$, after column-wise softmax of the matching score matrix $\boldsymbol{M}$, the sum of probabilities of matching degree of $\boldsymbol{h_p^X}$ on reference answer $\boldsymbol{A}$ is still 1, so that the model with the general attention mechanism cannot effectively distinguish the invalid sentences in the student answer. We utilize mutual attention mechanism to solve this problem.

Row-wise softmax function is applied to $\boldsymbol{M}$ to obtain the one-way attention matrix $\boldsymbol{\beta}$ of the reference answer to the student answer, where $\boldsymbol{\beta} \in R^{n\times \mathrm{m}}$. For sentence $\boldsymbol{h}_q^A$ in reference answer $\boldsymbol{A}$, $\boldsymbol{\beta}(q)$ represents the probability distributions of matching degree between $\boldsymbol{h}_q^A$ and each sentence in student answer $\boldsymbol{X}$.

$$\boldsymbol{\beta}(q) = softmax\left(M_{q,1}, \ldots, M_{q,m}\right)$$

$$\boldsymbol{\beta} = [\boldsymbol{\beta}(1), \ldots, \boldsymbol{\beta}(n)] \tag{5}$$

$\boldsymbol{\beta}$ is averaged on each column direction, and we get a weight vector $\boldsymbol{\beta_{ave}}$.

$$\boldsymbol{\beta_{ave}} = \frac{1}{n}\sum\nolimits_{q=1}^{n} \boldsymbol{\beta}(q) \tag{6}$$

$\boldsymbol{\beta_{ave}} = \left(\beta_{ave}^1, \beta_{ave}^2, \ldots, \beta_{ave}^m\right)$, where β_{ave}^m represents the matching weight of m th sentence in student answer to the whole reference answer. Next, we calculate the mutual attention-based weight vector $\boldsymbol{\alpha_{A\_X}}$ between the student answer and reference answer.

$$\boldsymbol{\alpha_{A\_X}} = \boldsymbol{\alpha} \cdot \boldsymbol{\beta}_{\boldsymbol{ave}}^T \tag{7}$$

$\boldsymbol{\alpha_{A\_X}} = \left(\alpha_{A\_X}^1, \alpha_{A\_X}^2, \ldots, \alpha_{A\_X}^n\right)$, where $\alpha_{A\_X}^n$ is the matching score of the overall student answer to n th sentence in reference answer. According to the mutual attention-based weight vector $\boldsymbol{\alpha_{A\_X}}$, the reference representation $\boldsymbol{A_X} = \left[\boldsymbol{h}_{\boldsymbol{1}}^{\boldsymbol{A_X}}, \boldsymbol{h}_{\boldsymbol{2}}^{\boldsymbol{A_X}}, \ldots, \boldsymbol{h}_{\boldsymbol{n}}^{\boldsymbol{A_X}}\right]$ is calculated specifically for student answer $\boldsymbol{X}$, which indicates the matching degree between student answer $\boldsymbol{X}$ and reference answer $\boldsymbol{A}$, where $\boldsymbol{A_X} \in R^{n\times 2d_1}$.

$$\boldsymbol{A_X} = \boldsymbol{A} \times \boldsymbol{\alpha_{A\_X}} \tag{8}$$

The Bi-LSTM Network for Text Feature Extraction. By using the semantic feature extraction network and mutual attention network, we obtain student answer $\boldsymbol{X}$ and reference answer $\boldsymbol{A_X}$. Sentences in $\boldsymbol{X}$ and $\boldsymbol{A_X}$ are respectively encoded by Bi-LSTM to capture the dependency between sentences. After the max-pooling over the hidden state vectors of all timesteps, we can get the encoded student answer vector $\boldsymbol{v_X}$ and reference answer vector $\boldsymbol{v_A}$, respectively.

$$\boldsymbol{v_X} = \mathrm{BiLSTM}(\boldsymbol{X}),\ \boldsymbol{v_A} = \mathrm{BiLSTM}(\boldsymbol{A_X}) \tag{9}$$

where $\boldsymbol{v_X} \in R^{2d_2}$, $\boldsymbol{v_A} \in R^{2d_2}$, d_2 is the number of Bi-LSTM hidden units.

Student Answer Scoring. The student answer vector $\boldsymbol{v_X}$ and the reference answer vector $\boldsymbol{v_A}$ are concatenated as the overall vector $\boldsymbol{v}$. Then $\boldsymbol{v}$ is fed into a two-layer feedforward neural network, and we can get the category c as the final score of the student answer through a softmax function.

$$\boldsymbol{v} = \boldsymbol{v_X} \oplus \boldsymbol{v_A}$$

$$\boldsymbol{v}_1 = relu(\boldsymbol{W_1} \cdot \boldsymbol{v} + \boldsymbol{b_1}), \boldsymbol{v}_2 = relu(\boldsymbol{W_2} \cdot \boldsymbol{v_1} + \boldsymbol{b_2})$$

$$c = softmax(\boldsymbol{v}_2) \tag{10}$$

We minimize the following cross entropy loss function when training the model.

$$L(\boldsymbol{\Theta}) = -\sum_{i=1}^{r} c_i \log p_{c_i} \tag{11}$$

Where r is the number of categories, $c_i \in \{0, 1\}$ is the real category of the sample, p_{c_i} is the probability that the sample is predicted to be category c_i, and $\boldsymbol{\Theta}$ is the set of all parameters in the model.

4 Experiments and Analysis

4.1 Datasets

We adopt the real datasets on a national specialty examination provided by our partner, which include student answers and expert reviews, as well as the reference answers. The dataset I contains 45,000 answers and scores range from 0, 1.5 and 3. The dataset II contains 40,000 student answers and scores range from 0, 1 and 1.5.

Each question is associated with a reading material on a specialty case. It requires the student to make a judgement according to the question and present his reasons. For example, the question "李某是否有权拒绝张某的赔偿请求?请简要说明理由 (Is Li's right to refuse Zhang's claim for compensation? Briefly explain the reason)". If a student makes a wrong judgement, he gets 0 point. If his judgement is correct but the reason is wrong, he gets 1.5 points. Only both his judgement and reason are correct, he gets 3 points. The statistics of datasets are shown in Table 1.

Table 1. The statistics of datasets.

Datasets	Full score	Number of student answers	Score categories and counts	
I	3	45000	0	8545
			1.5	10928
			3	25527
II	1.5	40000	0	5590
			1	18607
			1.5	15803

Each dataset is divided into training set, validation set and test set with the proportions 60%, 20%, and 20%, respectively. Taking into account the practical requirement on a small amount of samples, we also select the proportion 0.5%, 1%, 5%, 10% and 30% as training set, respectively. For comparison purpose, the verification set and test set remain 20%.

4.2 Comparison Models

Conv-GRNN. Conv-GRNN was proposed by Tang et al. [9]. The model first used CNN to encode sentence at lexical level, then generated a text vector through GRU at sentence level, and finally classified the text according to the text vector.

LSTM-GRNN. LSTM-GRNN was proposed by Tang et al. [9]. The model first used LSTM to encode sentences at lexical level, then generated a text vector through GRU at sentence level, and finally classified the text according to the text vector.

HN-AVE. HN-AVE was proposed by Yang et al. [16]. The model first used bidirectional GRU to encode sentence at lexical level, taking the mean of hidden state vectors of all timesteps as the sentence vector. Then the sentence vector sequence was input into another bidirectional GRU, taking the mean of hidden state vectors of all timesteps as the text vector and finally classified the text according to the text vector.

HN-MAX. HN-MAX was proposed by Yang et al. [16]. The model first encoded the sentence at lexical level using bidirectional GRU, taking the max-pooling result of the hidden state vectors of all timesteps as the sentence vector. Then the sentence vector sequence was input into another bidirectional GRU, taking the max-pooling result of hidden state vectors of all timesteps as the text vector and finally classified the text according to the text vector.
Our model and variants are as follows.

Bi-LSTM-CA-MAX. Bi-LSTM-CA-MAX is our model.

Bi-LSTM-CA-AVE. Bi-LSTM-CA-AVE is a variant of our model. The average of hidden state vectors of all timesteps of Bi-LSTM is taken as the output, and the other parts are the same as Bi-LSTM-MAX.

Bi-LSTM-CA. Bi-LSTM-CA is a variant of our model. The hidden state of the last timestep of Bi-LSTM is taken as the output, and the other parts are the same as Bi-LSTM-MAX.

Bi-LSTM-A. Bi-LSTM-A is a variant of our model. The general attention mechanism is used to calculate attention weights between student answer and reference answer. Each row of the one-way attention matrix $\boldsymbol{\alpha}$ is summed to obtain the final attention weights. The other parts are the same as Bi-LSTM-CA.

4.3 Experiment Setting and Metrics

Each student answer and reference answer are segmented into 20 sentences and 10 sentences, respectively, with zero vectors padded when the number of sentences was insufficient. Each sentence is segmented into 20 words, with zero vectors padded when

the number of words is insufficient. The dimension of word embedding is 300. The number of Bi-LSTM hidden units is set as 100. In Conv-GRNN, LSTM-GRNN, HN-AVE and HN-MAX, the student answer and the reference answer are input into the model, respectively. In Conv-GRNN and LSTM-GRNN, the outputs of GRU layer of the student answer and reference answer are concatenated as the input of next layer. In HN-AVE and HN-MAX, the outputs of the second GRU layer of the student answer and reference answer are concatenated as the input of next layer. The other parts were consistent with the original model.

We adopt the accuracy as the overall evaluation metric, with the precision P, recall R and F1 score on each category as metrics. The precision P_{c_i} on category c_i is the proportion of the number of samples classified to c_i whose real category is c_i to the total number of samples classified to c_i by the model. The recall R_{c_i} on c_i is the proportion of the number of samples classified to c_i by the model whose real category is c_i to the total number of samples whose real category is c_i.

4.4 Experimental Results Analysis

Accuracy Against Different Amount of Training Data. We first verify how much the amount of samples influence the performance. The comparison results are shown in Table 2. The Bi-LSTM-CA-MAX model outperforms other methods in most cases. The attention mechanism, such as in the models Bi-LSTM-A, Bi-LSTM-CA and Bi-LSTM-CA-MAX, contributes a lot on improving the performance, especially in the cases with less training data. For example, for training set 0.5%, the accuracy of our model increases by 2.2% and 2.5% on dataset I and dataset II comparing with non-attention models, respectively. For training set 1%, compared with models without attention mechanism, the accuracy of our models increases by 1.3% and 1.4% on dataset I and dataset II, respectively. We think the reason is that models without attention mechanism cannot capture the matching degree between the student answer and reference, so that the lack of training data has a greater limitation on the learning ability of these models. With the attention mechanism, we can use the matching information between the student answer and reference answer, and improve model performance when having less training data.

Table 2. Model comparison and evaluation against the training data ratio.

Models	DatasetI					DatasetII				
	0.5%	1%	5%	10%	30%	0.5%	1%	5%	10%	30%
Conv-GRNN	0.780	0.820	0.843	0.864	0.871	0.636	0.685	0.701	0.726	0.742
LSTM-GRNN	0.794	0.830	0.858	0.871	0.876	0.621	0.674	0.716	0.735	0.748
HN-AVE	0.826	0.841	0.869	0.878	0.882	0.675	0.715	0.731	0.749	0.751
HN-MAX	0.829	0.843	0.872	0.878	0.880	0.681	0.718	0.735	0.749	0.755
Bi-LSTM-A	0.831	0.848	0.876	0.876	0.881	0.680	0.719	0.739	0.742	0.754
Bi-LSTM-CA	0.838	0.849	0.878	0.876	**0.884**	0.695	0.725	0.743	0.75	0.756
Bi-LSTM-CA-MAX	**0.851**	**0.856**	**0.880**	**0.882**	0.883	**0.706**	**0.732**	**0.743**	**0.754**	0.756
Bi-LSTM-CA-AVE	0.845	0.850	0.879	0.879	0.881	0.700	0.725	0.741	0.751	**0.758**

As the amount of training data increases, the accuracy of each model grows lower. The gap of different models gradually becomes small, but our models are always at a leading position. Because the scope of answers to the question is relatively fixed, the changes of student answers are relatively small. Although comparison models cannot effectively utilize the matching information between the student answer and the reference answer, the results get better with the increasing size of data.

Analysis on Different Components of Models. Then we verify the performance of different network structures, pooling strategies and attention mechanisms, and get three conclusions according to the experimental results in Table 2.

The mutual attention mechanism is superior to the general attention mechanism. The performance of models with mutual attention mechanism, such as Bi-LSTM-CA, Bi-LSTM-CA-MAX and Bi-LSTM-CA-AVE, is better than Bi-LSTM-A with general attention mechanism. For training set 0.5%, Bi-LSTM-CA-MAX has an accuracy of 2.0% and 2.6% higher than Bi-LSTM-A on dataset I and dataset II, respectively. For training set 1%, the accuracy of Bi-LSTM-CA-MAX is 0.8% and 1.3% higher than Bi-LSTM-A on dataset I and dataset II respectively. The reason is that the mutual attention mechanism can capture the matching degree between student answer and reference answer more effectively than the general attention mechanism.

Models with the max-pooling perform better than models with the average-pooling. Compared with the models using average-pooling, the models with max-pooling achieved a better performance in terms of the overall accuracy. For training set 0.5% of dataset I, Bi-LSTM-CA-MAX and HN-MAX improved the accuracy by 0.6% and 0.3% than Bi-LSTM-CA-AVE and HN-AVE respectively. This might be caused by that the max-pooling strategy weakens the interference caused by the padding of zero vector compared with the average-pooling.

LSTM is superior to CNN. Based on the overall experimental results, LSTM-GRNN achieved a better performance than Conv-GRNN. This may be caused by that CNN structure in Conv-GRNN ignores the long-distance dependence features between words when encoding sentence vectors, and loses lots of semantic information, which cannot handle the problem of misjudgment caused by similarity in lexical and phrase level between student answer and reference answer. LSTM-GRNN captured the long-distance dependency between words through the first LSTM layer, and retained more semantic information than CNN, which can overcome the above errors to some extent.

Analysis of Model Performance on Different Categories. In order to further analyze the performance difference of each model on each category, we calculate the recall and F1 score of each categories on dataset I, as shown in Fig. 2. Figures (a) and (b) show F1 score of each model with 0.5% and 30% of training data, respectively. Figures (c) and (d) show the recall of each model with 0.5% and 30% of training data, respectively.

For training set 0.5%, the overall performance of our model is significantly better than comparison models in terms of F1 score and recall, which indicates the outstanding ability of our model in the situation of less training data. As the amount of training data increases to 30%, the gap between models gradually narrows, but the overall performance of our model is still in a leading position.

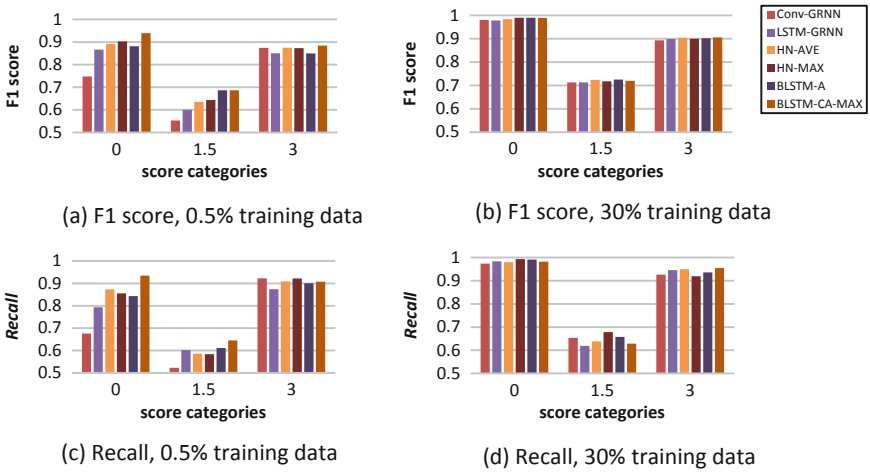

Fig. 2. Models comparison on dataset I.

The recall and F1 score of our model are significantly high than comparison models on the category of 0-point, such as Conv-GRNN, when training set is 0.5%. This indicates our model is good at capturing the student's judgement to the question, especially when the training data is insufficient.

It is notable that the recall and F1 score of 1.5-points category are significantly lower than the other two categories. The improvement of models on 1.5-points category is less than other categories as the training data increases. This is caused by the characteristics of datasets mentioned in previous section. The student answers might hit key words or phrases in reference answer, but are with wrong reason. The model cannot distinguish them and classify the student answer to 3-points category wrongly, which resulting in the low recall and F1 score of 1.5-points category.

5 Conclusion

For the task of grading Chinese answers on specialty subjective questions with reference answers, we propose a grading model which captures the matching degree between student answer and reference through Bi-LSTM network and mutual attention mechanism. We verify our model on real datasets of a national specialty examination against different amount of training samples, and analyze the performance of different network structures, pooling strategies and attention mechanisms. The experimental results demonstrate the effectiveness of our method. In the future, we are planning to investigate how to extract the knowledge points related to the specialty question from the textbook, then utilize the specialty knowledge to solve this grading task and answer the specialty question automatically.

Acknowledgments. This work was supported by the National Key R&D Program of China (Grant No. 2018YFC0831401), the National Natural Science Foundation of China (Grant No. 91646119), the Major Project of NSF Shandong Province (Grant No. ZR2018ZB0420), and the Key Research and Development Program of Shandong province (Grant No. 2017GGX10114). The scientific calculations in this paper have been done on the HPC Cloud Platform of Shandong University.

References

1. Burstein, J.: The E-rater® scoring engine: automated essay scoring with natural language processing. Shermis, M.D., Burstein, J.C. (eds.), pp. 113–121 (2003)
2. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27, Annual Conference on Neural Information Processing Systems, pp. 3104–3112. MIT Press, Montreal (2014)
3. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
4. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751. ACL, Doha (2014)
5. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems 28, Annual Conference on Neural Information Processing Systems, pp. 649–657. ACL, Montreal (2015)
6. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 655–665. ACL, Baltimore (2014)
7. Schwenk, H., Barrault, L., Conneau, A., LeCun, Y.: Very deep convolutional networks for text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, pp. 1107–1116. ACL, Valencia (2017)
8. Johnson, R., Zhang, T.: Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 562–570. ACL, Vancouver (2017)
9. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1422–1432. ACL, Lisbon (2015)
10. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2267–2273. AAAI, Austin (2015)
11. Shi, Y., Yao, K., Tian, L., Jiang, D.: Deep LSTM based feature mapping for query classification. In: The 2016 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 1501–1511. NAACL, San Diego (2016)
12. Xiao, Y., Cho, K.: Efficient character-level text classification by combining convolution and recurrent layers. arXiv:1602.00367 (2016)
13. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations. San Diego (2015)
14. Tan, M., Xiang, B., Zhou, B.: LSTM-based deep learning models for non-factoid answer selection. arXiv:1511.04108 (2015)

15. Chaturvedi, A., Pandit, O.A., Garain, U.: CNN for text-based multiple choice question answering. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 272–277. ACL, Melbourne (2018)
16. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for text classification. In: The 2016 Conference of the North American Chapter of the Association for Computational Linguistics, pp. 1480–1489. NAACL, San Diego (2016)
17. Cui, Y., Liu, T., Chen, Z., Wang, S., Hu, G.: Consensus attention-based neural networks for Chinese reading comprehension. In: 26th International Conference on Computational Linguistics, pp. 1777–1786. ACM, Osaka (2016)
18. Cui, Y., Chen, Z., Wei, S., Wang, S., Liu, T., Hu, G.: Attention-over-attention neural networks for reading comprehension. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 593–602. ACL, Vancouver (2017)
19. Li, S., Zhao, Z., Hu, R., Li, W., Liu, T., Du, X.: Analogical reasoning on Chinese morphological and semantic relations. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 138–143. ACL, Melbourne (2018)