Automated Essay Scoring via Pairwise Contrastive Regression

Jiayi Xie*, Kaiwei Cai*, Li Kong, Junsheng Zhou†, Weiguang Qu

NLP Lab, Department of Computer and Electronic Information
Nanjing Normal University, China
{ tammyjiayixie, ckwcccwkc }@gmail.com
kongli@nnu.edu.cn, { zhoujs, wgqu }@njnu.edu.cn

Abstract

Automated essay scoring (AES) involves the prediction of a score relating to the writing quality of an essay. Most existing works in AES utilize regression objectives or ranking objectives respectively. However, the two types of methods are highly complementary. To this end, in this paper we take inspiration from contrastive learning and propose a novel unified Neural Pairwise Contrastive Regression (NPCR) model in which both objectives are optimized simultaneously as a single loss. Specifically, we first design a neural pairwise ranking model to guarantee the global ranking order in a large list of essays, and then we further extend this pairwise ranking model to predict the relative scores between an input essay and several reference essays. Additionally, a multi-sample voting strategy is employed for inference. We use Quadratic Weighted Kappa to evaluate our model on the public Automated Student Assessment Prize (ASAP) dataset, and the experimental results demonstrate that NPCR outperforms previous methods by a large margin, achieving the state-of-the-art average performance for the AES task[1].

1 Introduction

Automated Essay Scoring (AES) is to evaluate the quality of essays and score automatically by using computer technologies. Notably, reasonable grading can solve problems that consume much time and require a lot of human effort. What's more, providing feedback to learners can promote self improvement. It is one of the most important applications of natural language processing (NLP) and is widely required in the educational field.

Most existing methods typically recast AES as a regression task, where the goal is to predict the score of an essay (Taghipour and Ng, 2016; Dong and Zhang, 2016; Dong et al., 2017; Tay et al., 2018). Although some promising results have been achieved, these regression-based models cannot exploit the labelling information in the training data efficiently and directly. Besides, another line of research treats AES as a preference ranking problem with learning-to-rank methods. Yannakoudakis et al. (2011) first proposed to rank the pair of documents by extracting features; later, Chen and He (2013) transformed this task into a listwise ranking problem. Cummins et al. (2016) also performed transfer learning to rank two essays that are constrained to be from the same prompt.

However, most existing works in AES utilize regression objectives or ranking objectives respectively. As a matter of fact, the two types of methods are highly complementary. On the one hand, only using regression models for AES cannot explicitly model score relationships between essays in the training data. On the other hand, only using ranking-based models could not guarantee accurate scores. In effect, in real-life situations, when a teacher evaluates and grades a student's essay, he usually first compares it with one or multiple exemplar essays as reference and then gives a specific score for it.

Recently, Yang et al. (2020) presents the first work to combine regression and ranking in the AES task by applying a multi-loss method that optimizes regression loss and ranking loss jointly with a simple dynamic combination strategy. Nevertheless, it is actually quite difficult to determine the combination weights to achieve the tradeoff between the two optimization objectives. Additionally, the proposed batch-wise learning based ranking model sacrifices the accuracy and only ranks essays in each batch.

To address the above problems, in this paper we explore a unified framework for the AES task where both the regression objective and the ranking

* Equal contribution.
† Corresponding author.
[1]The source code is available at https://github.com/CarryCKW/AES-NPCR.

Proceedings of the 29th International Conference on Computational Linguistics, pages 2724–2733
October 12–17, 2022.

objective are optimized simultaneously, with the goal of incorporating the merits of two popular AES solutions. The key challenge here is how to integrate two significantly different optimization objectives into a single model with a single loss.

To this end, we take inspiration from contrastive learning (Yu et al., 2021; Chen et al., 2020) and propose a novel unified Neural Pairwise Contrastive Regression (NPCR) model that jointly optimizes the two objectives in a principled way. In a nutshell, the goal of contrastive learning is to learn a better representation space (Chen et al., 2020). In particular, for two given essays, the distance between similar essays from the same category should be small while the distance between dissimilar essays should be large, and the semantic relationship can be reflected by measuring the distance in the representation space. Thus, under the contrastive learning framework, the proposed model aims to map the input essays into the representation space and calculates the differences between essays by the relative scores. Specifically, we first design a neural pairwise ranking model to guarantee the global ranking order in a large list of essays, and then we further extend this neural pairwise ranking model to predict the relative scores between an input essay and several reference essays. Additionally, a multi-sample voting strategy is adopted for the inference for every input test essay.

We use Quadratic Weighted Kappa to evaluate our model on the Automated Student Assessment Prize (ASAP) dataset, and the experimental results demonstrate that the proposed model outperforms previous methods by a large margin and establishes new state-of-the-art on this public benchmark.

In summary, the contributions of this work can be concluded as follows: (1) To the best of our knowledge, we make the first attempt to explore a unified framework for the AES task that performs regression and ranking optimization simultaneously; (2) We propose a neural pairwise ranking model for AES that guarantees the global ranking order in a large list of essays; (3) Experimental results on the public dataset ASAP show that the proposed approach not only achieves the state-of-the-art average performance but also obtains better performance on almost all prompts compared to all baselines.

2 Background

2.1 Task Description

Automated essay scoring systems are used in evaluating and scoring student essays written based on a given prompt. The performance of these systems is assessed by comparing their scores assigned to a set of essays to human-assigned gold standard scores. Since the output of AES systems is usually a real-valued number, the task is often addressed as a supervised machine learning task (mostly by regression or preference ranking).

2.2 The Multi-loss Method for the Combination of Regression and Ranking

In order to take advantage of the complementarity of regression loss and ranking loss, Yang et al. (2020) proposes a multi-loss objective to fine-tune the BERT model for the AES task by using a simple dynamic optimizing strategy as Formula 1. However, it is very difficult to determine the suitable combination weights to achieve the tradeoff between the two losses.

$$L = \tau_e \times L_m + (1 - \tau_e) \times L_r \quad (1)$$

where L_m is the regression objective, L_r is the result of the batchwise loss function, and τ_e is the parameter that vary with the number of epoch.

Besides, Yang et al. (2020) uses a batch-wise approach ListNet which ranks a list of essays each time and measures the accuracy between the predicted ranking list and the ground truth label. The major defect of this method is that it can only rank essays in a batch and cannot guarantee precise global order.

3 Pairwise Contrastive Regression for AES

3.1 Methodology Overview

Traditionally, most existing works formulate AES as a regression task, where the input is an essay and the output is a predicted score relating to the writing quality of the essay (Taghipour and Ng, 2016; Dong and Zhang, 2016; Tay et al., 2018). Formally, given the input essay e with the score label s, the regression problem is to predict the score $\hat{s}$ based on the quality of input essay:

$$\hat{s} = R_\theta(F_W(e)) \quad (2)$$

where R_θ and F_W are the regressor model and the feature extractor parameterized by θ and W, respectively.

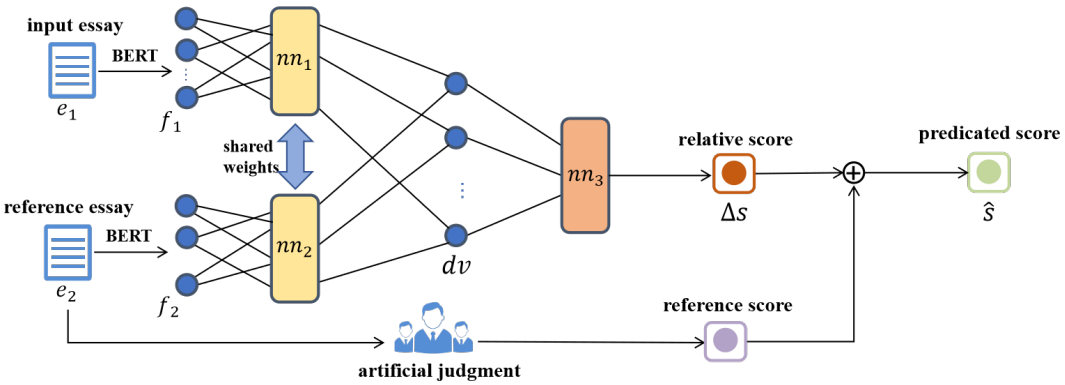

Figure 1: The overall framework of neural pairwise contrastive regression model for AES.

However, optimizing the regression objective alone is inadequate to make good use of the score label information in the training data. In contrast, the ranking-based methods could explicitly model score relationships between essays (Yannakoudakis et al., 2011). In order to take advantage of the complementarity of these two types of methods, we therefore propose to reformulate the AES problem as regressing relative score between the input and an exemplar. Let e_i denotes the input essay, and e_j denotes the reference essay with score label s_j, this regression problem can be re-written as:

$$\hat{s}_i = R_\theta(F_W(e_i, e_j)) + s_j \tag{3}$$

Note that the aim here is to predict the relative score, i.e. the difference of the scores between the input essay and a reference essay.

Technically speaking, the major challenge of successfully predicting the relative score lies in how to design effective regressor that takes as input a pair of essays rather than a single essay. In contrast to the single essay input, this regressor with the essay pair input should satisfy more characteristics, such as reflexivity and antisymmetry.

To achieve this, we propose a neural pairwise contrastive regression model for AES to predict the relative score. The overall framework of our method is illustrated in Figure 1. Methodologically, our pairwise contrastive regression model is actually a natural extension to a neural pairwise ranker for AES.

In order to clearly articulate our approach, in the following subsections, we first introduce the design of a neural pairwise ranker for AES in detail, and then we further extend it to a pairwise contrastive regressor to reach this goal.

3.2 Neural Pairwise Learning to Rank for AES

In this section, motivated by the DirectRanker (Köppel et al., 2019), we aim to design a neural pairwise ranking model for AES to predict a global ranking order given a large list of essays. To do this, given any two essays e_1 and e_2, we first define a partial order operator $e_1 \succeq e_2$ such that e_1 has higher score than e_2. In order to achieve a consistent and global order, this operator should satisfy three characteristics: *reflexivity*, *antisymmetry* and *transitivity*. Further, we use a ranking function $rf : \mathcal{F} \times \mathcal{F} \rightarrow \mathcal{R}$ over the feature space $\mathcal{F}$ to implement the operator:

$$x \succeq y :\Leftrightarrow rf(x, y) \geq 0, \ for\ x, y \in \mathcal{F} \tag{4}$$

Thus, the three characteristics of this operator can be defined through the function rf as follows:

(A) Reflexivity: $rf(x, x) = 0$

(B) Antisymmetry: $rf(x, y) = -rf(y, x)$

(C) Transitivity: $(rf(x, y) \geq 0 \wedge rf(y, z) \geq 0) \Rightarrow rf(x, z) \geq 0$

Particularly, to meet these requirements, the ranking function rf can be implemented by using a neural network with specific structure.

Firstly, in order to map an input essay e to low-dimensional vector space, we use BERT (Devlin et al., 2019) which can make full use of rich semantic information to obtain the text vector representation f:

$$h = BERT(e) \ \in \mathcal{R}^{r_h * |e|} \tag{5}$$

$$f = h_{[CLS]} \tag{6}$$

where h is the hidden representations, r_h is the dimension of the hidden state and $|e|$ represents the length of the input essay. The vector f, a hidden representation mapping to the special token $[CLS]$, is used as the text representation for the input essay e.

Next, as shown in Figure 1, the feature extraction part of the model includes two subnets nn_1 and nn_2 which are composed of multi-layer perceptron. The two subnets share the same structure and parameters like weights, biases, activation, etc. Then a difference vector of the two outputs from two subnets nn_1 and nn_2 can be simply calculated as follows:

$$dv = F_w(f_1) - F_w(f_2) \tag{7}$$

where f_1 and f_2 are the representation vectors of one essay pair, and F_w is the feature extractor parameterized by w.

After that, the difference vector dv is fed into the third subnet nn_3 which has only one output neuron. As shown by (Köppel et al., 2019), the antisymmetry can easily be guaranteed by choosing antisymmetric activation functions and removing the biases of the neuron.

In fact, it is easy to prove that the above three characteristics can be satisfied in our model. More specifically, we first utilize ϕ to define the antisymmetric activation function, i.e. $\phi(-x) = -\phi(x)$ for $\phi : \mathcal{R} \to \mathcal{R}$.

(I) The satisfaction of (II) means that (I) can be inferred.

$$rf(x, x) = -rf(x, x) \Rightarrow rf(x, x) \equiv 0 \tag{8}$$

(II) From the above mentioned, nn_1 and nn_2 have the consistent network structure, thus they employ the same function $g : \mathcal{F} \to \mathcal{R}^n$. Hence, for two input feature vectors $x, y \in \mathcal{F}$, (B) can be proved as follows:

$$\begin{aligned} rf(x, y) &= \phi[w(g(x) - g(y))] \\ &= -\phi[wg(y) - wg(x)] \\ &= -rf(y, x) \end{aligned} \tag{9}$$

where w is the weight vector.

(III) Assuming $x, y, z \in \mathcal{F}$, $rf(x, y) \geq 0$ and $rf(y, z) \geq 0$, the transitivity of the model can be testified by:

$$\begin{aligned} rf(x, z) &= \phi[w(g(x) - g(z))] \\ &= \phi[wg(x) - wg(y) + wg(y) - wg(z)] \\ &= rf(x, y) + rf(y, z) \geq 0 \end{aligned} \tag{10}$$

where w is the weight vector and g is defined as in (II). Hence, (C) is compliant.

3.3 Pairwise Contrastive Regression Model for AES

Next, we extend the neural pairwise ranking model illustrated in the previous section to form a pairwise contrastive regression model that predicts the relative score between an input essay and an reference essay. In fact, it is relatively straightforward to achieve this.

As the pairwise ranking model, the output of the third subnet nn_3 shown in Figure 1 is just a binary value. If we allow the output value of this subset to be a real value corresponding to a relative score and specify the second input essay as the reference sample, the converted model is actually a basic contrastive regression model. Then, the difference vector dv is fed into a fully connected neural network which consists of only one output neuron with antisymmetric activation and without a bias. Given the difference vector dv, the pairwise contrastive model can be simply defined as follows:

$$\Delta s = R_\theta(dv) \tag{11}$$

where Δs represents the relative score of any two essays and θ is the parameter of the regression model. This regression problem of the relative scores can be solved by minimizing the Mean Squared Error (MSE) loss that can be computed based on the predicted relative scores and the golden relative scores over the training data. The corresponding loss function for this pairwise contrastive regression is shown as follows:

$$L_r = \frac{1}{N}\sum_{i=1}^{N}(\Delta s_i - \Delta s_i^{'})^2 \tag{12}$$

where N refers to the total number of essay pairs, Δs_i and $\Delta s_i^{'}$ denote the predicted relative score and the golden relative score, respectively.

In principle, this contrastive regression model should satisfy three characteristics: *reflexivity*, *antisymmetry* and *accumulation*. Specifically, the *accumulation* can be defined as follows:

$$rf(x, y) + rf(y, z) = rf(x, z) \tag{13}$$

Obviously, both *reflexivity* and *antisymmetry* can easily be satisfied by adopting the same neural network architecture as in the previous section. Nevertheless, in theory, the *accumulation* in this contrastive regression model cannot be guaranteed by any neural network model itself. Thus, a learning goal of this contrastive regression model is to meet accumulation as much as possible.

With this end in view, how to select the essays pairs during training becomes critical. Therefore we design an effective selection strategy for constructing the training data. Particularly, we first arrange all essays in each prompt as a sequence according to the order in which they appear in the training data, then orderly pick every two adjacent essays in the sequence as a pair. Furthermore, an important additional step is imposed to cater the need of accumulation. To be specific, if we pick the essays pairs (e_i, e_j) and (e_j, e_k) as the training instances, we should also add the pair (e_i, e_k) into the training data in order to make the learned model meet the accumulation. Additionally, to make the input essay and the reference essay comparable, we tend to select the essays that shares the same prompt with the input essay as the references.

3.4 Inference

During inference, we employ a multi-sample voting strategy. Intuitively, the selected reference samples should be comparable to the input test essays. However, our dataset has eight prompts and different prompts have different relative score ranges. In order to solve the above problem, we select some sample essays which have the same prompt with the input essays.

Specifically, given an input essay e_{test}, we select M samples from the training datasets to construct M pairs using these M different samples $\{e^m_{train}\}^M_{m=1}$ whose scores are $\{s_{train}\}^M_{m=1}$. Then we will obtain M predicted scores and the final score of the input essay is the average of these M scores. The process of multi-sample voting can be summarized as follows:

$$\hat{s}^m_{test} = R_\theta(F_w(e_{test}, e^m_{train})) + s^m_{train} \quad (14)$$

$$\hat{s}_{test} = \frac{1}{M}\sum_{m=1}^{M} \hat{s}^m_{test},\ m = 1, 2, ..., M \quad (15)$$

where θ and w are the parameters of the pairwise contrastive regression model. $\hat{s}^m_{test}$ represents the m-th predicted score and $\hat{s}_{test}$ denotes the final predicted score of the input essay e_{test}.

Prompt ID	Essay Set Size	Original Score Range	Relative Score Range
1	1783	2-12	-10-10
2	1800	1-6	-5-5
3	1726	0-3	-3-3
4	1772	0-3	-3-3
5	1805	0-4	-4-4
6	1800	0-4	-4-4
7	1569	0-30	-30-30
8	723	0-60	-60-60

Table 1: The details of the ASAP dataset.

4 Experiments

4.1 Experimental Settings

4.1.1 Dataset

We use the widely used dataset ASAP (Automated Student Assessment Prize) for experimental evaluation. This comes from the competition which was organized and sponsored by the William and Flora Hewlett Foundation (Hewlett). This dataset contains eight prompts and has different genres and different number of essays, as described in Table 1. Following previous work, we also utilize 5-fold cross-validation to evaluate the model. In each run, we use 60%, 20% and 20% of the dataset for each prompt as training data, validation data and test set, which are provided by (Taghipour and Ng, 2016).

4.1.2 Evaluation Metrics

In this paper, we use the commonly used metric Quadratic Weighted Kappa (QWK) to measure the agreement between the artificial scores and the predicted results. Specially, let the essay set be scored on a scale of 1 to N, and the score from the expert is i while the predicted score of the model is j.

$$K = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}} \quad (16)$$

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2} \quad (17)$$

where w, O, E are matrices of weights, observed scores and expected scores, respectively. Furthermore, the value of $O_{i,j}$ represents the number of essays that receive a score i by the human rater and a score j by the AES system. And $E_{i,j}$ represents the outer product between two histogram vectors of the scores.

Dataset/Prompts									
Model	1	2	3	4	5	6	7	8	Avg
EASE(SVR)	0.781	0.630	0.621	0.749	0.782	0.771	0.727	0.534	0.699
EASE(BLRR)	0.761	0.621	0.606	0.742	0.784	0.775	0.730	0.617	0.705
ALL-MTL-cTAP (2016)	0.816	0.667	0.654	0.783	0.801	0.778	0.787	0.692	0.747
CNN+LSTM (2016)	0.821	0.688	0.694	0.805	0.807	0.819	0.808	0.644	0.761
LSTM-CNN-attent (2017)	0.822	0.682	0.672	0.814	0.803	0.811	0.801	0.705	0.764
SKIPFLOW (2018)	0.832	0.684	0.695	0.788	0.815	0.810	0.800	0.697	0.764
HISK+BOSWE (2018)	0.845	0.729	0.684	0.829	0.833	0.830	0.804	0.729	0.785
R^2BERT (2020)	0.817	0.719	0.698	0.845	0.841	0.847	**0.839**	0.744	0.794
NPCR	**0.856**	**0.750**	**0.756**	**0.851**	**0.847**	**0.858**	0.838	**0.779**	**0.817**

Table 2: The QWK evaluation scores on ASAP dataset, and the results of baselines are adapted from their original papers.

4.1.3 Implementation Details

Following previous work (Yang et al., 2020), we also use $BERT_{base}$ model for fair comparison. For tokenization and vocabulary, and we all use the preprocessing tools provided by the BERT model. For the limitation of our GPU memory, we set the max length of the essay is 512 words and the batch size is 5. We train our model for 80 epochs and select the best model based on the performance on the validation set. We use AdamW as our optimizer to train the model and the initial learning rate is set to $1e-5$. In addition, we normalize all relative scores to the range of $[0, 1]$ during training and the scores are rescaled back to the original score range for evaluation. Following previous work, we conduct the evaluation in prompt-specific fashion.

4.2 Overall Performance

In this section, we comprehensively compare our overall performance with the following state-of-the-art related methods that were evaluated on the dataset ASAP.

4.2.1 Baselines

EASE The major non deep learning system that we compare against is the Enhanced AI Scoring Engine (EASE). This system is publicly available and also achieved excellent results in the ASAP competition. Following previous works, we report the results of EASE with the settings of Support Vector Regression (SVR) and Bayesian Linear Ridge Regression (BLRR).

ALL-MTL-cTAP Cummins et al. (2016) used a constrained multi-task pairwise-preference learning method to achieve the representation of the essays.

CNN+LSTM Taghipour and Ng (2016) first designed a neural network model which used CNN for word sequence modeling and LSTM for text level modeling. Then the essay representation is achieved by mean of time pooling.

LSTM-CNN-attent Dong et al. (2017) proposed to use hierarchical neural networks with attention mechanism to extract features from sentences and documents.

SKIPFLOW Tay et al. (2018) proposed the model that considered neural coherence features within the context of an end-to-end neural framework to improve prediction.

HISK+BOSWE Cozma et al. (2018) combined string kernels and word embeddings to extract more semantic features and gained higher performance in both in-domain and cross-domain settings.

R^2BERT Yang et al. (2020) presented the first work that employed a multi-loss method to combine regression and ranking and to fine-tune BERT models in AES tasks.

4.2.2 Performance Comparison

Table 2 shows the overall performance comparison between our model and the above state-of-the-art AES models. From Table 2, we can see that our approach substantially improves the average QWK score by 2.3%, compared to the best baseline R^2BERT. It is worth noting that our model not only achieves the state-of-the-art average performance but also obtains better performance on almost all prompts compared to all baselines, which shows the superiority of the proposed pairwise contrastive regression model for AES.

Model	Avg QWK
R^2BERT-RegrOnly	0.768
NPCR-RegrOnly	0.770
R^2BERT-RankOnly	0.756
NPCR-RankOnly	0.796
NPCR	**0.817**

Table 3: Ablation studies on the use of the regression and ranking objectives in our model.

Model	Avg QWK
NPCR-Accu	0.800
NPCR-Group	0.802
NPCR	**0.817**

Table 4: Performance comparison on the choice of reference essays during training and inference.

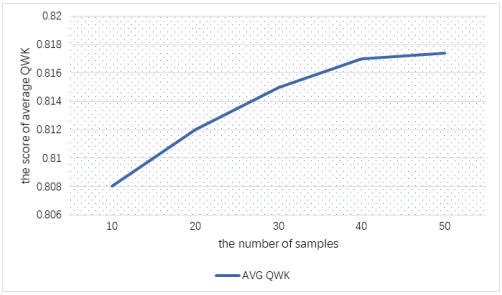

Figure 2: The performance curve varying with different number of reference essays.

4.3 Analysis

4.3.1 Effect of Contrastive Regression Learning

Unlike previous regression-based models for AES, our approach integrates regression with ranking within a contrastive regression framework. In this section, we evaluate the effect of exploiting contrastive regression by the ablation test.

As shown in the first row of Table 3, two baselines are presented for comparison. The first baseline R^2BERT-RegrOnly refers to the regression only version of R^2BERT (Yang et al., 2020). On the other hand, we also implement the second baseline NPCR-RegrOnly, which is the regression only version of our model NPCR by removing the contrastive learning from NPCR. More specifically, we first use BERT to obtain the representations of the input essays and then employ a fully connected layer with a sigmoid activation function to predict the scores. The results in Table 3 show that, compared the two baselines, our full model NPCR consistently improve QWK scores by 4.9% and by 4.7% respectively, which clearly indicates the importance of contrastive regression learning for our model.

4.3.2 Effect of Pairwise Ranking

In this section, we inspect the effect of our neural pairwise ranking model. In the second row of Table 3, the first baseline R^2BERT-RankOnly refers to the ranking only version of R^2BERT (Yang et al., 2020). Similarly, we also implement the second baseline NPCR-RankOnly, which is the ranking only version of our model NPCR by removing the score prediction part from NPCR. In detail, the output label of NPCR-RankOnly is a binary value which represents the priority relationship between any two essays. During inference, NPCR-RankOnly does not apply the multi-sample voting strategy, which means M is set to 1. After observing the results in Table 3, we can infer the following two implications: Firstly, the performance of NPCR-RankOnly is 4.0% better than R^2BERT-RankOnly, which shows that our neural pairwise ranking method is superior to the neural batchwise based ranking model in previous (Yang et al., 2020). Secondly, the large gap between our full model NPCR and the baseline NPCR-RankOnly clearly demonstrates the complementarity of the two methods.

4.3.3 Effect of the Strategy of Choosing Training Sample Pairs

In order to meet the accumulation of our model NPCR, we propose a sample selection strategy for building an effective training dataset, as illustrated in Section 3.3. In this section, we inspect the effect of this sample selection strategy. As a comparison, we also implement a baseline NPCR-Accu, which does not consider the accumulation while choosing the training sample pairs. That is to say, only the two adjacent essays in the given essay sequence are added into the training dataset. The results in Table 4 show that the average QWK score of NPCR is 1.7% better than the baseline NPCR-Accu.

4.3.4 Effect of the Strategy of Choosing Reference Essays

It is necessary to choose the number M in multi-sample voting strategy for inference. In this section, we investigate the relationship between the prediction performance and the number of reference essays. Figure 2 shows the results predicted with different number of reference essays. The per-

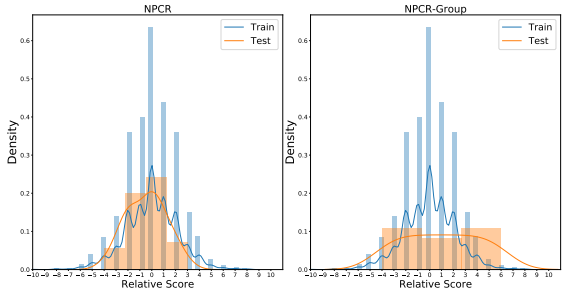

Figure 3: Realtive Score Distribution Histogram and Gaussian Kernel Density Estimation for model NPCR and NPCR-Group in Prompt 1.

Model	Avg QWK
NPCR-XLNet	0.816
NPCR-BERT	0.817
NPCR-RoBERTa	0.817

Table 5: Performance comparison of different pre-trained language models.

formance curve in Figure 2 demonstrates that the performance gradually improves as the number M increases, and then the performance growth tends to converge when M is greater than 40.

Furthermore, we consider the impact of different sample scores' distribution and devise a group-testing strategy to verify it. Concretely, we first divide the score range of training essays into M non-overlapping intervals (called $'groups'$), and then select M reference essays by picking only one essay from every group. From Table 4 we can see that NPCR has better performance than NPCR-Group that uses the group-testing strategy, indicating that randomly choosing samples is better than selecting samples from different score groups. For detailed reason, we generate Relative Socres Distribution Histograms and observe the Gaussian Kernel Density Estimates for the training and test essay pairs in all datasets, for instance Prompt 1 in Figure 3, in order to study the characteristics of the data distribution under different strategies of choosing reference essays. We can find that NPCR has better consistency w.r.t the distributions of relative scores in training and test data than NPCR-Group.

4.3.5 Comparison of Different Pre-trained Language Models

In this section we investigate the performance variation with different mainstream pre-trained language models including BERT, XLNet and RoBERTa. The experimental results in Table 5 show that the performance remains almost constant when changing the underlying pre-trained language model in our approach, which indicates that our AES model NPCR under the contrastive regression learning framework is relatively insensitive to the choice of pre-trained language models.

4.3.6 Computational Cost

In this section, we analyze the computational cost of the model NPCR. Compared with the previous work dealing with a single essay, our model NPCR really needs to take slightly more computational cost. However, in our model NPCR, the runtime of dealing with an essay pair is roughly similar to the cost of dealing with a single essay in the baselines, thus leading to the limitation of the increase of the computational cost. Hence, the number of essay pairs is a critical factor for analyzing the computational cost. In summary, during training, if the number of essays in training dataset is n, the number of essay pairs to be calculated in NPCR is less than $2*n$; during inference, if the number of essays in the test set is n, the number of essay pairs needed to be checked in NPCR should be $M*n$, where M is the number of reference essays. In effect, when we record the running time of model NPCR in Prompt 1 with GPU RTX3090Ti, the average training runtime is 90 seconds per epoch and the average inference runtime is 0.7 second per 40 essay pairs.

5 Related Work

Automated essay scoring systems have been deployed for high-stakes assessment since decades ago. The early approaches for AES mainly involved handcrafted feature based methods (Larkey, 1998; Chodorow and Burstein, 2004; Phandi et al., 2015; Zesch et al., 2015), while the recent studies have explored deep learning based methods to deliver state-of-the-art performance for this task.

In recent years, the mainstreams of AES methods typically formulate AES as a regression task. Multiple deep learning architectures based regression models for AES have been proposed. Taghipour and Ng (2016) presents the first neural network model for AES, which first uses the combination of CNN and LSTM to extract features of essays to generate text representation vectors and then apply a linear layer with sigmoid activation to map the vectors to valid scores. Dong and Zhang (2016) uses a hierarchical structure to automatically learn features from the word level and the sentence level.

Dong et al. (2017) further introduces the attention mechanism into the model and proves that CNN is more conducive to obtaining local features, while LSTM is more suitable for obtaining global features. Tay et al. (2018) proposes to consider neural coherence features as auxiliary features for prediction within an end-to-end neural framework. Recent advances in BERT (Devlin et al., 2019) model have inspired researchers to use pre-trained language model in AES (Rodriguez et al., 2019; Mim et al., 2019; Song et al., 2020).

Another line of research focuses on applying the learning to rank methods in AES tasks. Yannakoudakis et al. (2011) firstly formulate AES as a rank preference problem and then employ a pairwise ranking model RankSVM to rank two or more essays based on statistical features. Chen and He (2013) further utilizes the listwise ranking method to learn a ranking model based on linguistic features. Cummins et al. (2016) uses multi-task learning to address the problem of prompt adaptation by treating each prompt as a different task and introducing a constrained preference-ranking approach.

Recently, considering the complementarity of ranking and regression approaches, Yang et al. (2020) proposes a multi-loss method to combine regression and ranking in the AES task with a simple dynamic combination strategy.

6 Conclusion

In this paper, aiming to incorporate the merits of two popular AES solutions, we propose a novel unified model NPCR for AES which combines both regression and ranking objective in a principled way. Our approach is conceptually simple, however, the experimental results on the public dataset ASAP demonstrate that NPCR significantly outperforms previous approaches, advancing the state of the art in AES tasks.

In future work, we will explore more sophisticated neural feature extractors under the pairwise contrastive regression framework so that more powerful text features can be learned from the input essays, such as the hierarchical structure of a document, coherence features and so on.

References

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1741–1752.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 1597–1607.

Martin Chodorow and Jill Burstein. 2004. Beyond essay length: evaluating e-rater®'s performance on toefl® essays. *ETS Research Report Series*.

Mădălina Cozma, Andrei M Butnaru, and Radu Tudor Ionescu. 2018. Automated essay scoring with string kernels and word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 503–509.

Ronan Cummins, Meng Zhang, and Edward Briscoe. 2016. Constrained multi-task learning for automated essay scoring. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186.

Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring–an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1072–1077.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL)*, pages 153–162.

Marius Köppel, Alexander Segner, Martin Wagener, Lukas Pensel, Andreas Karwath, and Stefan Kramer. 2019. Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance. In *Machine Learning and Knowledge Discovery in Databases- European Conference (ECML, PKDD)*, pages 237–252.

Leah S Larkey. 1998. Automatic essay grading using text categorization techniques. In *SIGIR '98: Proceedings of the 21st Annual International (ACM,SIGIR)*, pages 90–95.

Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi, and Kentaro Inui. 2019. Unsupervised learning of discourse-aware text representation for essay scoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 378–385.

Peter Phandi, Kian Ming A Chai, and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 431–439.

Pedro Uria Rodriguez, Amir Jafari, and Christopher M Ormerod. 2019. Language models and automated essay scoring. *CoRR*, abs/1909.09482.

Wei Song, Kai Zhang, Ruiji Fu, Lizhen Liu, Ting Liu, and Miaomiao Cheng. 2020. Multi-stage pre-training for automated chinese essay scoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6723–6733.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in NaturalLanguage Processing (EMNLP)*, pages 1882–1891.

Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui. 2018. Skipflow: Incorporating neural coherence features for end-to-end automatic text scoring. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 5948–5955.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics (EMNLP)*, pages 1560–1569.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 180–189.

Xumin Yu, Yongming Rao, Wenliang Zhao, Jiwen Lu, and Jie Zhou. 2021. Group-aware contrastive regression for action quality assessment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7919–7928.

Torsten Zesch, Michael Wojatzki, and Dirk Scholten-Akoun. 2015. Task-independent features for automated essay grading. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications (BEA@NAACL-HLT)*, pages 224–232.