Automatic Essay Scoring Incorporating Rating Schema via Reinforcement Learning

Yucheng Wang[1], **Zhongyu Wei**[23], **Yaqian Zhou**[13]*, **Xuanjing Huang**[13]
[1]School of Computer Science, Fudan University, Shanghai, China
[2]School of Data Science, Fudan University, Shanghai, China
[3]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China
{yuchengwang14,zywei,zhouyaqian,xjhuang}@fudan.edu.cn

Abstract

Automatic essay scoring (AES) is the task of assigning grades to essays without human interference. Existing systems for AES are typically trained to predict the score of each single essay at a time without considering the rating schema. In order to address this issue, we propose a reinforcement learning framework for essay scoring that incorporates quadratic weighted kappa as guidance to optimize the scoring system. Experiment results on benchmark datasets show the effectiveness of our framework.

1 Introduction

In recent years, neural networks have been widely used to grade student essays automatically and achieve state-of-the-art performance. In particular, a distributed representation is learned for an essay with variant neural networks and a linear layer is then used to produce the final score. Existing researches focus on learning better essay representation using different neural networks, including long short-term memory (LSTM) network (Taghipour and Ng, 2016), hierarchical convolutional neural networks (CNN) (Dong and Zhang, 2016), hierarchical CNN-LSTM structure with attention mechanism (Dong et al., 2017), and SKIPFLOW LSTM (Tay et al., 2017).

The major evaluation metric for AES is quadratic weighted kappa (QWK), which is also the official metric of Automated Student Assessment Prize[1] (ASAP). It evaluates the scoring results by taking rating schema into consideration. Because QWK is not differentiable, it is hard to train systems via optimizing this metric directly. Alternatively, existing AES systems are typically trained to predict the score for a single essay and optimized using mean square error (MSE). The gap between training and testing also limits the performance of state-of-the-art AES systems.

Recently, reinforcement learning (RL) has been introduced to optimize models in terms of non-differentiable quality metrics and studies have shown its effectiveness for various tasks including language generation (Ranzato et al., 2015; Rennie et al., 2016; Zhang et al., 2017), machine translation (Bahdanau et al., 2016) and relation classification (Feng et al., 2018).

Inspired by these researches, we propose a novel reinforcement learning framework that incorporates QWK as the guidance to optimize the essay scoring system. In our framework, we score a pack of essays at a time and the scoring of each single essay is treated as an action. The QWK value computed for the pack of essays is then delivered as a reward to update the scoring system. Because the existing regression-based essay scorer is unable to generate a probability distribution in nature, it is non-trivial to be used within the reinforcement learning framework. We therefore propose to use a classification-based scoring system instead. The proposed framework is evaluated in the benchmark datasets from ASAP and experiment results confirm its effectiveness on two different settings of essay representation structures.

2 Model

Typically, an essay scorer contains two components, namely, essay representation and essay scoring. The component of essay representation transforms an input essay into a distributed vector and the component of essay scoring assigns a score to the essay based on the vector. Both components are usually trained jointly. In order to incorporate QWK to guide the process of essay scoring, we introduce a novel essay scoring strategy named *packed evaluation*. At each time, essay scorer grades a pack of essays together with the target essay, and QWK is calculated for the pack.

*Corresponding author
[1]https://www.kaggle.com/c/asap-aes

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 791–797
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

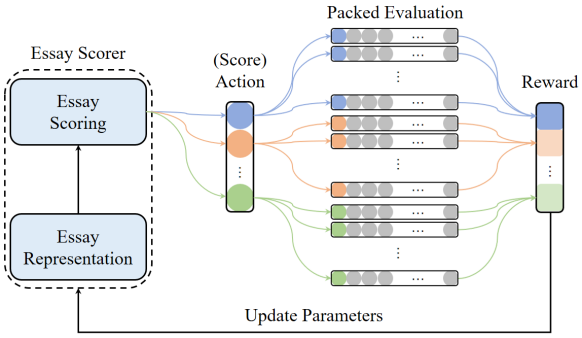

Figure 1: The reinforcement learning framework for automatic essay scoring. Node in color stands for a target essay and nodes in grey are essays randomly chose to form a pack for QWK calculation.

To avoid contingency, for each target essay, we repeat the packed evaluation multiple times by randomly choosing other essays in a pack. And the average QWK it achieves is set to be the reward. The reward is then delivered to the essay scorer as a weak signal to supervise the scorer. Figure 1 illustrates the training process of our model. We will introduce the different parts in detail in the rest of this section.

2.1 Essay Representation

This component converts an input essay into a dense vector as its representation. Recurrent neural networks (Williams and Zipser, 1989) are widely used to learn a representation for a sequence of words for essay scoring. Following existing researches, we also use recurrent neural networks (RNN) and test two different structures.

Bidirectional LSTM We first use a double-layer bi-directional LSTM network (Hochreiter and Schmidhuber, 1997) to process the essay. LSTM is a variant of recurrent neural network which uses gates to control the information flow. Our LSTM processes one word at a timestamp. Given the word embedding sequence $\{x_1, x_2, ..., x_n\}$ for the essay, the hidden states of the LSTM are calculated as follows:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$

$$h_t = o_t \circ tanh(c_t)$$

where $W_f, W_i, W_o, W_c, U_f, U_i, U_o$ and U_c are weight matrices, b_f, b_i, b_o and b_c are bias vectors. σ denotes sigmoid function and $\circ$ denotes element-wise multiplication.

In particular, the average value over all hidden states of each LSTM layer are computed, and we concatenate the mean states of the two layers together as the embedding vector of the essay. Given $h_{i,j}$ as the *j-th* hidden state of the *i-th* layer, the layer outputs and the essay embedding vector E are defined as follows:

$$output_i = \frac{1}{n}\sum_{j=1}^{n} h_{i,j}$$

$$E = \begin{bmatrix} output_0 \\ output_1 \end{bmatrix}$$

Dilated LSTM Dilated recurrent neural networks (Chang et al., 2017) are proved to be more effective than traditional RNNs in long sequence processing, by capturing multi-timescale information along the sequence, with the mechanism of dilated skip connections.

Denoting $s_t^i = f(x_t^i, s_{t-1}^i)$ as the iteration of cell states in traditional RNNs, states in dilated RNNs are iterated as $s_t^i = f(x_t^i, s_{t-k^i}^i)$, where k^i is the skip length in the *i-th* layer. In order to keep the most information active, we simply concatenate the average hidden states of every layer to form the essay embedding.

$$E = concat(output_i), for\ i\ in\ 1, 2, ..., L$$

where L is the number of layers.

2.2 Essay Scoring

Traditionally, a linear layer with sigmoid function is used to score an essay. Given an essay embedding vector E, the essay score is calculated as follows:

$$y = sigmoid(W_l^T E + b_l)$$

where W_l and b_l are weight vector and bias for scoring. By running n examples together, mean square error is used to evaluate the predicted score.

$$loss_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where y and $\hat{y}$ are score vectors representing predicted scores and ground truth scores, respectively. As we can see, such objective function is unable to take rating schema into consideration.

The regression-based scorer only outputs a single value without probability distribution. It is thus non-trivial to use it for policy learning in RL framework directly. Therefore, we propose to use a classification-based scorer, in which different score categories and their probabilities constitute an action space.

Classification-based Scoring We first feed the essay vector into a fully connected layer, then softmax function is used to transform the output into a probability distribution. Given an essay embedding vector E, the probability distribution vector c is calculated as follows:

$$c = softmax(W_c E + b_c)$$

where W_c and b_c are weight matrix and bias vector, respectively.

Given the ground truth category, cross entropy loss is applied to evaluate the agreement of the probabilities as follows:

$$loss_{CE} = -\sum_{i=1}^{N} Y_i log(c_i)$$

where N is the number of categories, which is equal to the number of possible ratings. Y is a one-hot vector with the element representing the ground truth category set as one.

Inter-class Penalty Cross entropy loss used in classification-based scorer does not imply the difference between categories, i.e. the rank information that is deemed to be important for essay scoring. Thus we enforce a penalty in addition to the cross entropy loss. Inspired by the definition of QWK, the penalty vector p is defined as follows:

$$p_i = \frac{(i - score)^2}{(N-1)^2}$$

where $score$ is the ground truth score of the essay.

The penalty loss function is defined as:

$$loss_P = \sum_{i=1}^{N} c_i p_i$$

Mixed Scoring In practice, we jointly train both a regression-based scoring layer and a classification-based scoring layer over the same document representation to help the classification-based scorer converge. By combining the two scorers together, the overall loss function can be written as:

$$loss_{pre} = \alpha_0 loss_{MSE} + \beta_0 loss_{CE} + \gamma_0 loss_P$$

where α_0, β_0 and γ_0 are hyper parameters. Mixed scoring is used as a pre-train model for our essay scorer in the phase of reinforcement learning.

2.3 Reinforcement Learning

We define our loss function as the negative expected reward:

$$loss_{RL} = -\mathrm{E}_{\tau \sim p(\tau)} r(\tau)$$

where τ is the set of actions, r denotes the reward, which is the average QWK an essay achieves in the packed evaluation.

By running n examples at a time, according to the REINFORCE algorithm (Williams, 1992), an approximated gradient can be calculated by:

$$\frac{\partial loss_{RL}}{\partial \theta} = \sum_{i=1}^{n} [\partial log(p_{i,y}|E_i;\theta) R_i]$$

where θ denotes all parameters relevant to score calculation, and $\partial log(p_{i,y}|E_i;\theta)$ can be computed by standard back propagation.

Note that only the classification-based scorer is involved in the process of reinforcement learning for essay scoring. The overall loss function for this phase can be written as:

$$loss_{overall} = \alpha_1 loss_{RL} + \beta_1 loss_{CE} + \gamma_1 loss_P$$

where α_1, β_1 and γ_1 are hyper parameters.

2.4 Quadratic Weighted Kappa(QWK)

QWK calculation emphasizes on the overall rating schema. By setting QWK as the reward, our model is trained at a macro aspect taking the grading specialty of different sets of essays into consideration.

An *N-by-N* quadratic weight matrix *W* is first computed to encode the rating information.

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

where N is the number of possible ratings. An *N-by-N* matrix *A* is calculated such that $A_{i,j}$ corresponds to the number of essays that receive a score *i* by the human rater, and a score *j* by the scoring system. Another *N-by-N* matrix *B* is constructed as the outer product of the histogram vectors of the two ratings. *A* and *B* are then normalized such that they have the same sum. Finally, from the three matrices, the quadratic weighted kappa is calculated as follows:

$$\kappa = 1 - \frac{\Sigma_{i,j} W_{i,j} A_{i,j}}{\Sigma_{i,j} W_{i,j} B_{i,j}}$$

set	# of essays	avg. of len.	rating range
1	1783	350	2-12
2	1800	350	1-6
3	1726	150	0-3
4	1772	150	0-3
5	1805	150	0-4
6	1800	150	0-4
7	1569	250	0-30
8	723	650	0-60

Table 1: Details of the ASAP dataset.

set	B0	B1	P0	RL0	P1	RL1
1	.711	.666	.666	.680	.759	**.766**
2	.582	.579	.579	.589	.630	**.659**
3	.627	.653	.662	.670	.673	**.688**
4	.758	.758	.761	.771	.768	**.778**
5	.768	.781	.786	.796	.795	**.805**
6	.776	.786	.787	**.798**	.790	.791
7	**.766**	.711	.726	.737	.748	.760
8	**.604**	.491	.525	.545	.536	.545
Avg	.699	.678	.687	.698	.712	**.724**

Table 2: Experiment results for different models in terms of QWK. **Bolded** number is the best performance in each row.

3 Experiment

3.1 Experiment Setup

The ASAP dataset is used for evaluation. It consists of essays written by middle-school English-speaking students ranging among eight different topics. More details are listed in Table 1. As there are no released labels for the test data, we separate the validation set and test set from the original training data. Following Taghipour and Ng (2016) and Dong et al. (2017), we use 5-fold cross-validation. In each fold, the split is 60%, 20%, 20% for training, validation and testing respectively.

All essays are parsed with the NLTK[2] tokenizer. We pre-train the word embedding via word2vec (Mikolov et al., 2013) on the whole dataset. The number of hidden states in LSTMs is 200. We use a four-layer double-directional dilated LSTM with skip lengths 1,2,4,8 in each layer respectively. During the training and the scoring, scores are scaled to range [0,1] for regression-based scorer. They are restored to integers when calculating QWK values. In the RL phase, the pack size is 64 essays, and packed evaluation is repeated 7 times per essay. The essay scorer for RL is pre-trained by mixed scoring.

We compare the performance of different approaches:

- **B0**: This model uses a double-layer bi-directional LSTM to encode an essay and mean square error as objective function to train the essay scorer;
- **B1**: This is a classification-based scorer and it is trained jointly with a regression-based scorer;
- **P0**: Based on B1, this model incorporates penalty loss function;
- **P1**: This model shares the same settings with *P0*, but uses dilated LSTM instead of a double-layer bi-directional LSTM for essay representation;
- **RL0**: This model uses *P0* as the scorer under our reinforcement learning framework;
- **RL1**: This model uses *P1* as the scorer under our reinforcement learning framework.

[2]http://www.nltk.org

3.2 Results

The overall results of our models in terms of QWK are shown in Table 2. We have the following findings:

- By incorporating a penalty loss to the classification scorer, the performance of *P0* is equal to or better than *B1* on all the eight sets. This indicates the effectiveness of combining rank information with cross-entropy loss for essay scoring.
- By replacing double-layer bi-directional LSTM with dilated LSTM, *P1* improves the QWK values by a large margin compared with *P0* on all the eight sets. This indicates the effectiveness of using dilated LSTM for document representation for the task of automatic essay scoring. The performance improvement brought by *P1* compared to *P0* is even greater when the length of essays are higher (set 1,2,7,8, see Table 1), indicating that dilated networks are specifically better at long sequence processing.
- By incorporating QWK to guide the optimization of essay scorer, approaches (*RL0* and *RL1*) with reinforcement learning strategy can improve the performance consistently on all the eight sets compared to their counterparts (*P0* and *P1*). We also performed

one-tailed t-test, showing that the improvements brought by reinforcement learning are significant with $p < 0.05$ compared to their base scorer models (*RL1* vs. *P1* and *RL2* vs. *P2*).

- The performance of classification-based scorer *B1* can equate or improve the performance on four datasets (set 3,4,5,6) compared with regression-based scorer *B0*. The rating ranges for set 1,2,7,8 are much greater than set 3,4,5,6 (see Table 1). The performance difference between *B1* and *B0* decreases (from positive to negative) when the number of rating categories increases. This is because when the number of categories get larger, it requires much more parameters for the classification-based scorer to be well trained. Given N categories, the classification layer should output N probabilities for each category per essay, costing N times more parameters than regression-based scoring.

4 Related Work

There are two lines of research related to our work including text quality evaluation and reinforcement learning for natural language processing.

4.1 Text Quality Evaluation

Traditionally, AES models are usually divided into three categories: classification, regression and ranking. Naive Bayes models are mostly used in classification tasks. Larkey (1998) use bag-of-word features. Following that, Rudner and Liang (2002) develop a system based on multinomial Bernoulli Naive Bayes, using content and style features. E-rater (Attali and Burstein, 2004) is one of the earliest systems to adopt regression methods. Phandi et al. (2015) use correlated Bayesian Linear Ridge Regression (cBLRR) focusing on domain-adaptation tasks. Ranking models use linguistic features. Yannakoudakis et al. (2011) formulate AES as a pair-wise ranking problem by ranking the order of pair essays. Chen and He (2013) formulate AES into a list-wise ranking problem by considering the order relation among the whole essays.

Argument quality evaluation is a task closely related to AES, which involves evaluation of argumentative texts with various grains (argument-level, post-level, etc.). Tan et al. (2016); Wei et al. (2016a); Wang et al. (2017) make use of linguistic features to evaluate the persuasiveness of arguments in online forums. Wei et al. (2016b); Ji et al. (2018) consider features from the perspectives of argumentation interaction between participants. Persing and Ng (2017) construct their model based on error types for argumentation.

4.2 Reinforcement Learning for Natural Language Processing

Being able to optimize non-differentiable quality metrics, reinforcement learning has been widely used in natural language processing tasks such as machine translation (Bahdanau et al., 2016), image captioning (Rennie et al., 2016; Zhang et al., 2017) and text summarization (Ranzato et al., 2015). To the best of our knowledge, this paper is the first attempt to optimize the scorer by QWK that considers rating schema.

Skip connections in RNNs are capable of capturing long-term dependencies in sequences. Vezhnevets et al. (2017) introduces dilated LSTM to allow its manager to operate at a low temporal resolution. Yu et al. (2017) propose a reinforcement learning method to let the network learn how long to skip.

5 Conclusion and Future Work

In this paper, we propose a reinforcement learning framework incorporating QWK metric as the reward to train the essay scoring system directly. A packed evaluation strategy is used for QWK computation and the scoring of each essay is treated as a single action. In particular, dilated LSTM is used to encode an essay, and a softmax layer is utilized for essay grading. Experiment results on benchmark datasets prove that training the grading system toward QWK is effective.

Further analysis on experiment results indicates the disadvantage of using a classification-based scorer for essays with complex grading schema. One of the future directions can be exploring other kinds of scoring actions than classification under the reinforcement learning framework.

Acknowledgments

Thanks for the constructive comments from anonymous reviewers. This work is partially supported by National Natural Science Foundation of China (Grant No. 61702106), Shanghai Science and Technology Commission (Grant No. 17JC1420200, Grant No. 17YF1427600 and Grant No. 16JC1420401).

References

Yigal Attali and Jill Burstein. 2004. Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series*, 2004(2).

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A Hasegawa-Johnson, and Thomas S Huang. 2017. Dilated recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 76–86.

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*, pages 1741–1752.

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*, 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 2017)*, pages 153–162.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of AAAI*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Lu Ji, Zhongyu Wei, Xiangkun Hu, Yang Liu, Qi Zhang, and Xuanjing Huang. 2018. Incorporating argument-level interactions for persuasion comments evaluation using co-attention model. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3703–3714.

Leah S Larkey. 1998. Automatic essay grading using text categorization techniques. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 90–95.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Isaac Persing and Vincent Ng. 2017. Why cant you convince me? modeling weaknesses in unpersuasive arguments. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4082–4088. AAAI Press.

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*, pages 431–439.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2016. Self-critical sequence training for image captioning. *arXiv preprint arXiv:1612.00563*.

Lawrence M Rudner and Tahung Liang. 2002. Automated essay scoring using bayes' theorem. *The Journal of Technology, Learning and Assessment*, 1(2).

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

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th international conference on world wide web*, pages 613–624. International World Wide Web Conferences Steering Committee.

Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui. 2017. Skipflow: Incorporating neural coherence features for end-to-end automatic text scoring. *arXiv preprint arXiv:1711.04981*.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. 2017. Feudal networks for hierarchical reinforcement learning. *arXiv preprint arXiv:1703.01161*.

Lu Wang, Nick Beauchamp, Sarah Shugars, and Kechen Qin. 2017. Winning on the merits: The joint effects of content and style on debate outcomes. *arXiv preprint arXiv:1705.05040*.

Zhongyu Wei, Yang Liu, and Yi Li. 2016a. Is this post persuasive? ranking argumentative comments in online forum. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 195–200.

Zhongyu Wei, Yandi Xia, Chen Li, Yang Liu, Zachary Stallbohm, Yi Li, and Yang Jin. 2016b. A preliminary study of disputation behavior in online debating

forum. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 166–171.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 180–189. Association for Computational Linguistics.

Adams Wei Yu, Hongrae Lee, and Quoc V Le. 2017. Learning to skim text. *arXiv preprint arXiv:1704.06877*.

Li Zhang, Flood Sung, Feng Liu, Tao Xiang, Shaogang Gong, Yongxin Yang, and Timothy M Hospedales. 2017. Actor-critic sequence training for image captioning. *arXiv preprint arXiv:1706.09601*.