




Automated Essay Scoring via Example-Based Learning

Yupin Yang  and Jiang Zhong 

Chongqing University, Chongqing 400044, China
{yyp, zhongjiang}@cqu.edu.cn

Abstract. Automated essay scoring (AES) is the task of assigning grades to essays. It can be applied for quality assessment as well as pricing on User Generated Content. Previous works mainly consider using the prompt information for scoring. However, some prompts are highly abstract, making it hard to score the essay only based on the relevance between the essay and the prompt. To solve the problem, we design an auxiliary task, where a dynamic semantic matching block is introduced to capture the hidden features with example-based learning. Besides, we provide a hierarchical model that can extract semantic features at both sentence-level and document-level. The weighted combination of the scores is obtained from the features above to get holistic scoring. Experimental results show that our model achieves higher Quadratic Weighted Kappa (QWK) scores on five of the eight prompts compared with previous methods on the ASAP dataset, which demonstrate the effectiveness of our model.

Keywords: Automated essay scoring · Natural language processing · Example-based learning

1 Introduction

Automated essay scoring (AES) is the task of employing computer programs to assign grades to essays based on their content, grammar, and structure. It has become an important educational application of natural language processing (NLP). For example, Educational Testing Service (ETS) uses AES systems to evaluate the writing ability of students. Such systems can also be applied for quality assessment as well as pricing on User Generated Content. Typically, AES systems regard the task as a regression problem based on handcrafted features (e.g., length-based features and lexical features) and most of them have achieved good results [1, 10, 16]. However, such systems require feature engineering, which costs lots of time and effort. Therefore, a large number of researchers focus on neural networks that are capable of modeling complex patterns without human assistance [3, 6, 11, 14].

Previous works mainly focus on the text itself [6, 13, 14], ignoring to investigate the topic information of the essays with prompts. Prompts indicate the

requirements and topics for students' writing. As is observed, essays off the prompt always receive low scores while high score essays are relevant to the prompt. Chen and Li [2] extracted the similarity of the essay with the topic on document-level for scoring and achieved good performance. But only using document-level features for scoring may lose some information in detail. To learn how each part of the essay sticks to the prompt more accurately, Zhang and Litman [17] proposed the Co-Attention Based Neural Network to model the similarity of essays at sentence level. However, some prompts are highly abstract, making it hard to score the essay only based on the similarity between the essay and the prompt. Thus, we introduce the example-based learning as auxiliary task to capture the hidden features.

Our main contributions are as follows:

- We design a dynamic semantic matching block to capture the hidden features with example-based learning, which is an auxiliary task for AES.
- We provide a hierarchical model that can extract semantic features at both sentence-level and document-level, which are useful for evaluating coherence and relevance in the essays.
- Experimental results show that our model achieves higher Quadratic Weighted Kappa (QWK) scores on five of the eight prompts compared with previous methods on the ASAP dataset.

2 Related Work

Automated Essay Scoring (AES) systems have been deployed for assigning grades to essays since decades ago. The first AES system created in 1996 is Project Essay Grade which uses linguistic surface features [12]. Recent works mainly use neural networks for automated essay scoring. Dong and Zhang [5] employed a two-layer CNN model to learn sentence representations and essay representations. Differently, Taghipour and Ng [14] used LSTM in their model which effectively learned features for scoring. However, these works only focus on the essay itself, despite the relatedness of the essay to the topic.

High score essays always keep to the prompt closely. Some researchers consider the relevance of the essay to the given prompt for scoring since an essay cannot get a high score if it is not relevant to the prompt. There are many ways to compute the relevance of an essay to the prompt. Higgins et al. [8] extracted sentence features based on semantic similarity measures and the discourse structure, to capture breakdowns in coherence. Chen et al. [2] proposed hierarchical neural networks and used the similarity between the essay and topic as auxiliary information for scoring. All of them take prompt relevance into account as it is an important part of the guidelines. However, it is hard to do semantic matching with the prompt because the prompt is composed of abstract and general sentences. In our approach, we generate relevance features by performing semantic matching with the high score essays. The relevance features are used as auxiliary features for prediction.

3 Model

In this section, we describe the proposed hierarchical structured model named AES-SE, which contains three parts: 1) coherence modeling block, 2) relevance modeling block, 3) dynamic semantic matching block (Fig. 1).

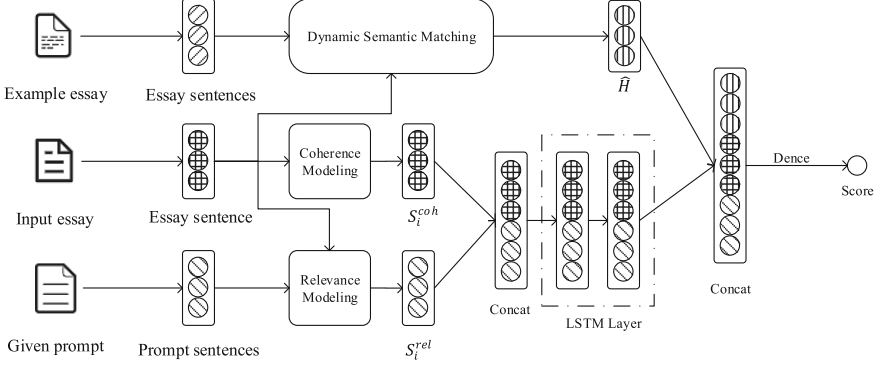


Fig. 1. An overview of our model. There are three parts: coherence modeling block, relevance modeling block and dynamic semantic matching block. All the extracted features are concatenated and sent to a dense layer for the final score.

3.1 Coherence and Relevance Modeling

For semantic coherence within a document and the relevance to the prompt, we apply the coherence modeling block and the relevance modeling block. It is not enough only considering features within cliques [7,9]. Instead, we use the self-attention mechanism to capture semantic changes within the whole document.

Sentence Representation. To capture lexical-semantic relations among words, we use pre-trained BERT [4] to get the sentence representation S_i .

$$S_i = BERT(W_e) \quad (1)$$

where W_e are the words of each sentence in the essay.

Coherence Modeling. To extract the coherence feature of the essay, we use self-attention mechanism to compute the similarity between sentences:

$$score(S_i, S_j) = S_i^T W_a S_j \quad (2)$$

where S_i and S_j are sentences from the essay $\{S_1, S_2, S_3, \dots, S_n\}$, W_a is the weight matrix to be learnt and the score function $score(S_i, S_j)$ tells how much similar the two sentences are.

$$\alpha_{ij} = \frac{\exp(score(S_i, S_j))}{\sum_{k=1}^n \exp(score(S_i, S_k))} \quad (3)$$

where α_{ij} represents the attention weight between S_i and other sentences.

$$S_i^{coh} = \sum_{j=1}^n \alpha_{ij} S_j \quad (4)$$

Finally, we use weighted sum of sentences as the coherence S_i^{coh} .

Relevance Modeling. It is observed that essays with high score always stick to the topic. To model the prompt relevance, we compute the similarity of essays with the assigned prompt. This process is almost the same as coherence modeling, where we compute the similarity between sentences from the essay and its prompt. The obtained relevance representation is S_i^{rel} .

3.2 Example-Based Learning

There are some consistent features that high-scoring essays usually have. Therefore, we design a dynamic semantic matching block to capture the hidden feature from high score essays as auxiliary information for holistic scoring.

Example Selection. To select typical examples, we use the k-means algorithm. We pick out full mark compositions, and use BERT to encode the sentences. Then, we take the averaged sentence vector of each essay as the input of k-means. Finally, we select essays that are closest to the cluster centers as examples.

Dynamic Semantic Matching. According to psychological researches, it is hard for people to pay close attention to too many things at the same time [15]. While understanding a text deeply, our focus may dynamically change to different sentences. With the aim to focus on the significant sentences with the consideration of learned information at each step, the dynamic semantic matching block is designed. To get the document representation of the essay. We utilize attention mechanism to integrate the sentences:

$$T_i = V_c \tanh(W_c S_i + b) \quad (5)$$

$$\gamma_i = \frac{\exp(T_i)}{\sum_{k=1}^n \exp(T_k)} \quad (6)$$

where γ_i is the attention weight. V_c , W_c , and b are parameters to be trained. The document representation h_e is weighted sum of sentence vector S .

$$h_e = \sum_{i=1}^n \gamma_i S_i \quad (7)$$

The same is done on the example essay to get the document representation h_s . The inputs of the dynamic semantic matching block are sentence vectors from input essay $T_e = \{S_1, S_2, S_3, \dots, S_n\}$ and the example essay $\{S'_1, S'_2, S'_3, \dots, S'_m\}$. For each step, an important sentence will be chosen for current input of an LSTM using attention mechanism. The choosing function $F_c(T_e, h_{t-1}, h_s)$ is formulated as follows:

$$Z_i = V_d^T \tanh(W_d S_i + U_d h_{t-1} + M_d h_s) \quad (8)$$

$$\delta_i = \frac{\exp(Z_i)}{\sum_{k=1}^n \exp(Z_k)} \quad (9)$$

$$\hat{a}_t = \sum_{i=1}^n \delta_i S_i \quad (10)$$

where V_d , W_d , U_d and M_d are parameters to be trained. h_s is the document representation of the example essay and h_{t-1} is the last step of the LSTM as follows:

$$\hat{h}_t = \text{LSTM}(\hat{a}_t, h_{t-1}) \quad (11)$$

We can get the last output \hat{h}_e from the LSTM where we compare the essay with the example. To compare the example to the essay, we can also get \hat{h}_s . Then, we send them to multi-layer perceptron (MLP) to calculate the relation probability R :

$$R = \text{MLP}(\hat{h}_e, \hat{h}_s, \hat{h}_e \odot \hat{h}_s, \hat{h}_e - \hat{h}_s) \quad (12)$$

where \odot means element-wise product. To each of the example essays, we repeat this process and get the averaged features \hat{H} :

$$\hat{H} = \frac{1}{q} \sum_{i=1}^q R_i \quad (13)$$

where q is the number of the example essays.

3.3 Scoring

After obtaining coherence features S^{coh} and relevance features S^{rel} , for each sentence, we concatenate the features together and send them to a BI-LSTM for modeling the document. After that, all the hidden states are fed into a mean-over-time layer. The function is defined as follows, where n denotes the num of sentences in an essay and h_t is the hidden state of the BI-LSTM at time t .

$$h_t = \text{BI-LSTM}(h_{t-1}, [S_t^{coh}, S_t^{rel}]) \quad (14)$$

$$H = \frac{1}{n} \sum_{t=1}^n h_t \quad (15)$$

Finally, we use the sigmoid function to compute the final score.

$$y = \sigma(W_y[H; \hat{H}] + b_y) \quad (16)$$

where W_y and b_y indicate the weight matrix and bias. H is the semantic representation of the essay. \hat{H} is the semantic matching feature.

As for loss function, we use mean squared error (MSE) [6]. MSE is used to compute the average value of squared error between the predicted scores and golden ones, as follows:

$$mse(y, y^*) = \frac{1}{N} \sum_{i=1}^N (y_i - y_i^*)^2 \quad (17)$$

where y is the predicted score and y^* is the true value.

4 Experiments

In this section, we introduce the dataset and evaluation metric we use and the experimental results.

4.1 Dataset

We use the ASAP (Automated Student Assessment Prize) dataset¹ as it has been widely used to evaluate the performance of AES systems. There are 12976 essays written by students with 8 prompts of different genres. The students were from Grade 7 to Grade 10 and 2 human graders scored the essays.

4.2 Evaluation Metric

Quadratic Weighted Kappa (QWK) is the official evaluation metric in the ASAP competition, which measures the agreement between ratings assigned by humans and ratings predicted by AES systems. As the ASAP dataset is used in this paper for evaluation, we adapt QWK as our evaluation metric.

4.3 Experimental Results

In this section, we test the performance of AES-SE and the baselines on the ASAP dataset. The results in Table 1 are the QWK scores on the eight prompts from the ASAP dataset, where the best results are bold. The baselines include RNN, GRU, LSTM, CNN, EASE, SKIPFLOW LSTM, and HISK+BOSWE+

¹ <https://www.kaggle.com/c/asap-aes/data>.

Table 1. Comparison with state-of-the-art methods on the ASAP dataset

Models	Prompt1	Prompt2	Prompt3	Prompt4	Prompt5	Prompt6	Prompt7	Prompt8	Average
RNN	0.687	0.633	0.552	0.744	0.744	0.757	0.743	0.553	0.675
GRU	0.616	0.591	0.668	0.787	0.795	0.800	0.752	0.573	0.698
EASE(SVR)	0.781	0.621	0.630	0.749	0.782	0.771	0.727	0.534	0.699
EASE(BLRR)	0.761	0.606	0.621	0.742	0.784	0.775	0.730	0.617	0.705
CNN	0.774	0.662	0.639	0.753	0.748	0.766	0.751	0.626	0.714
LSTM	0.780	0.697	0.683	0.787	0.795	0.767	0.758	0.651	0.740
SKIPFLOW LSTM	0.832	0.684	0.695	0.788	0.815	0.810	0.800	0.697	0.765
HISK+BOSWE and ν -SVR	0.845	0.729	0.684	0.829	0.833	0.830	0.804	0.729	0.785
AES-SE	0.864	0.727	0.717	0.823	0.838	0.835	0.812	0.694	0.788

ν -SVR, which achieved state-of-the-art performance on the ASAP dataset. Compared with HISK+BOSWE+ ν -SVR [3], AES-SE achieves higher QWK scores on five of the eight prompts and the average QWK score of AES-SE is also higher. As shown in Table 1, AES-SE achieves new state-of-the-art performance on five of the eight prompts and the averaged QWK score. On average of the eight prompts, our AES-SE achieves 0.788, which is 0.3% higher than HISK+BOSWE+ ν -SVR [3].

5 Conclusion

In this paper, we conduct a hierarchical structure named AES-SE with an auxiliary task for automated essay scoring. We use BERT to encode sentences capturing lexical-semantic relations among words. We simultaneously consider coherence features and relevance features to evaluate cohesion and task achievement. Moreover, with dynamic semantic matching block, the similarity of an essay with high score essays is computed as auxiliary information for scoring. Finally, we concatenate all the extracted features and compute the final score. Experimental results show that our model outperforms the current state-of-the-art methods with the improvement of the QWK score by 0.3%. In addition, we also achieve a significant 11.7% improvement over feature engineering baselines. For future work, we will explore using domain adaptation in our model.

Acknowledgment. This research was partially supported by the National Key Research and Development Program of China (2017YFB1402400 and 2017YFB1402401), the Key Research Program of Chongqing Science and Technology Bureau (cstc2020jscx-msxmX0149), the Key Research Program of Chongqing Science and Technology Bureau (cstc2019jscx-mbdxX0012), and the Key Research Program of Chongqing Science and Technology Bureau (cstc2019jscx-fxyd0142).

References

1. Attali, Y., Burstein, J.: Automated essay scoring with e-rater® v.2. J. Technol. Learn. Assess. 4(3) (2006)

2. Chen, M., Li, X.: Relevance-based automated essay scoring via hierarchical recurrent model. In: 2018 International Conference on Asian Language Processing (IALP), pp. 378–383. IEEE (2018)
3. Cozma, M., Butnaru, A.M., Ionescu, R.T.: Automated essay scoring with string kernels and word embeddings. In: ACL, no. 2, pp. 503–509 (2018). <https://aclanthology.info/papers/P18-2080/p18-2080>
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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, vol. 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis (2019). <https://doi.org/10.18653/v1/N19-1423>, <https://www.aclweb.org/anthology/N19-1423>
5. Dong, F., Zhang, Y.: Automatic features for essay scoring—an empirical study. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1072–1077 (2016)
6. Dong, F., Zhang, Y., Yang, J.: Attention-based recurrent convolutional neural network for automatic essay scoring. In: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pp. 153–162 (2017)
7. Farag, Y., Yannakoudakis, H., Briscoe, T.: Neural automated essay scoring and coherence modeling for adversarially crafted input. arXiv preprint [arXiv:1804.06898](https://arxiv.org/abs/1804.06898) (2018)
8. Higgins, D., Burstein, J., Marcu, D., Gentile, C.: Evaluating multiple aspects of coherence in student essays. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL, vol. 2004, pp. 185–192 (2004)
9. Li, J., Hovy, E.: A model of coherence based on distributed sentence representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2039–2048 (2014)
10. Liu, J., Xu, Y., Zhu, Y.: Automated essay scoring based on two-stage learning. arXiv preprint [arXiv:1901.07744](https://arxiv.org/abs/1901.07744) (2019)
11. Mesgar, M., Strube, M.: A neural local coherence model for text quality assessment. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4328–4339 (2018)
12. Page, E.B.: The use of the computer in analyzing student essays. *Int. Rev. Educ.* **14**, 210–225 (1968)
13. Süzen, N., Gorban, A.N., Levesley, J., Mirkes, E.M.: Automatic short answer grading and feedback using text mining methods. *Procedia Comput. Sci.* **169**, 726–743 (2020)
14. Taghipour, K., Ng, H.T.: A neural approach to automated essay scoring. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1882–1891 (2016)
15. Wang, J., Chen, H.C., Radach, R., Inhoff, A.: *Reading Chinese Script: A Cognitive Analysis*. Psychology Press, London (1999)
16. Zesch, T., Wojatzki, M., Scholten-Akoun, D.: Task-independent features for automated essay grading. In: Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 224–232 (2015)
17. Zhang, H., Litman, D.: Co-attention based neural network for source-dependent essay scoring. arXiv preprint [arXiv:1908.01993](https://arxiv.org/abs/1908.01993) (2019)