Automatic Comment Generation for Chinese Student Narrative Essays

Zhexin Zhang[1]*, Jian Guan[1]*, Guowei Xu[2], Yixiang Tian[2] and Minlie Huang[1]†

[1]The CoAI group, DCST; [1]Institute for Artificial Intelligence; [1]State Key Lab of Intelligent Technology and Systems;
[1]Beijing National Research Center for Information Science and Technology; [1]Tsinghua University, Beijing 100084, China.
[2]TAL Education Group.

{zx-zhang18,j-guan19}@mails.tsinghua.edu.cn, {xuguowei,tianyixiang}@tal.com
aihuang@tsinghua.edu.cn

Abstract

Automatic essay evaluation can help reduce teachers' workload and enable students to refine their works rapidly. Previous studies focus mainly on giving discrete scores for either the holistic quality or several distinct traits. However, real-world teachers usually provide detailed comments in natural language, which are more informative than single scores. In this paper, we present the comment generation task, which aims to generate comments for specified segments from given student narrative essays. To tackle this task, we propose a planning-based generation model, which first plans a sequence of keywords, and then expands these keywords into a complete comment. To improve the correctness and informativeness of generated comments, we adopt two following techniques: (1) training an error correction module to filter out incorrect keywords, and (2) recognizing fine-grained structured features from source essays to enrich the keywords. To support the evaluation of the task, we collect a human-written Chinese dataset, which contains 22,399 essay-comment pairs. Extensive experiments show that our model outperforms strong baselines significantly. Moreover, we exert explicit control on our model to generate comments to describe the strengths or weaknesses of inputs with a 91% success rate. We deploy the model at http://coai.cs.tsinghua.edu.cn/static/essayComment/. A demo video is available at https://youtu.be/IuFVk8dUxbI. Our code and data are available at https://github.com/thu-coai/EssayCommentGen.

1 Introduction

Automatic essay evaluation is a useful educational application of natural language processing (Page, 1966), which is beneficial for reducing teachers' workload and enabling students to improve writing skills independently. Prior studies focus mainly on automatic essay scoring (AES) in terms of either holistic scores (Cozma et al., 2018) or trait-specific scores (Mathias and Bhattacharyya, 2020; Song et al., 2020). However, real-world teachers usually provide detailed comments in natural language, which are more informative so that students can know more about the strengths and weaknesses of their works.

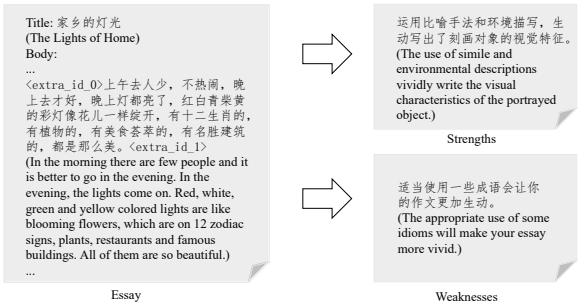

Figure 1: An example for the comment generation task. Given an essay with a specified segment (separated from the rest using special tokens <extra_id_0> and <extra_id_1>), the model should generate a sentence to comment the strengths or weaknesses of the segment.

In this work, we present the first study on automatic comment generation, which requires generating a fluent comment in natural language to describe strengths or weaknesses for a specified segment from a given student essay, as exemplified in Figure 1. We only focus on narrative essays in this work, which comprise more than 90% of our originally collected essays. The challenges of the task mainly lie in the following three folds: **(1)** Capturing the linguistic features of the essay, ranging from the wording, rhetorical methods (e.g., "simile" in the example) to discourse structures. **(2)** Generating coherent comments to correctly reflect the strengths or weaknesses (e.g., the segment does not use idioms in the example) of the essay. **(3)** Generating informative and diverse comments since generic comments such as "it is good" do not pro-

*Equal contribution.
†Corresponding author.

Proceedings of the The 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 214 - 223

vide any helpful guidance for students, and it is also expected to generate diverse comments for different essays.

To tackle the problem, we propose a planning-based generation model (Yao et al., 2019), which first plans a sequence of keywords concerning specific writing skills such as "simile", and then expands the keywords into a coherent comment. On the one hand, planning helps build explicit connections between essays and underlying skills, which alleviates the degeneration issue that the model focuses on predicting common elements such as "more vivid" and tends to generate generic comments (Fan et al., 2019). On the other hand, we can exert direct control on the intermediate keywords to improve the correctness and informativeness of generated comments. Specifically, we fine-tune BERT (Devlin et al., 2019) to serve as an error correction module to filter out incorrect keywords, which is trained to discriminate matching relations between essays and keywords. Moreover, we recognize structured features from source essays in terms of idioms, proverbs, quotes, descriptive and rhetorical methods using heuristic techniques or pretrained classifiers. Then we combine these features with the predicted keywords. To control the type of generated comments, we insert a binary control code before the keywords and comments (0/1 for describing strengths/weaknesses). In the comment generation stage, we inject noise into the ground-truth keywords during training (Tan et al., 2021) to alleviate the exposure bias issue introduced by planning (Ranzato et al., 2016).

To support training and evaluation of the proposed task, we collect a Chinese dataset that contains 22,399 essay-comment pairs. Extensive experiments show that our model outperforms strong baselines in correctness, informativeness and diversity. Furthermore, we build a website to enable real-time interaction with our deployed model, where a user can upload a Chinese student essay and see comments along with recognized structured features for most paragraphs.

2 Related Work

Automatic Essay Scoring There have been wide explorations for automatic essay scoring, including holistic essay scoring and trait-specific essay scoring (Mathias and Bhattacharyya, 2020). Holistic essay scoring aims to assign an overall score for the essay. Taghipour and Ng (2016); Tay et al. (2018) used LSTM and Dong and Zhang (2016); Dong et al. (2017) used CNN to give a total score for the essay. Cozma et al. (2018) utilized word embedding clusters and string kernels to achieve strong performances. Yang et al. (2020) jointly resolved the essay scoring task and the essay ranking task through fine-tuning the BERT model. Trait-specific essay scoring aims to assign different scores for different traits of an essay, such as thesis clarity (Ke et al., 2019), style (Mathias and Bhattacharyya, 2018) and narrative quality (Somasundaran et al., 2018). Mathias and Bhattacharyya (2020) compared different trait-agnostic approaches to automatically score many different essay traits. However, all these works assign numeric scores for an essay, while we focus on generating a readable comment.

Essay Assessment Systems Attali and Burstein (2006) constructed a system named E-rater, which could provide numeric scores for different features such as grammar and style. LinggleWrite (Tsai et al., 2020) focused on grammatical error correction and automatic essay scoring. The system most similar to ours is IFlyEA (Gong et al., 2021), which has grammar level analysis techniques and components for discourse and rhetoric analysis. It also integrates the fine-grained analysis to form a review for the whole essay using templates. However, our system is capable of generating diverse and natural comments without the usage of templates and could give comments for different segments of the essay.

Planning-based Generation Humans usually outline the overall framework before writing. Many works have explored planning-based text generation, which first predicts an intermediate representation as a plan and then generates the complete text conditioned on the plan. The plan could be a series of keywords (Yao et al., 2019), an action sequence (Fan et al., 2019; Goldfarb-Tarrant et al., 2020) or a dense keyword distribution (Kang and Hovy, 2020; Kong et al., 2021). Tan et al. (2021) progressively refined the produced domain-specific content keywords into complete passages in multiple stages. In this paper, we adapt planning to the automatic comment generation task and improve the correctness and informativeness by revising the intermediate keywords.

3 Method

We formulate our task as follows: given an essay $X = (x_1, x_2, \cdots, x_M)$ with M tokens and a specified segment from x_i to x_j (the segment is separated from the rest using special tokens), the model should generate a comment $Y = (y_1, y_2, \cdots, y_N)$ with N tokens for the segment. The comment either shows praise for the strengths of the segment, or gives advice to improve the weaknesses of the segment.

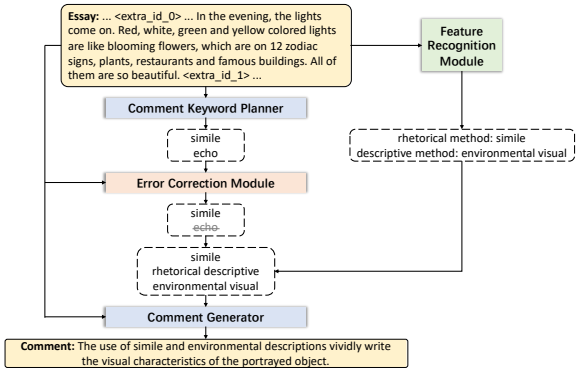

Figure 2: An overview of our model. The comment keyword planner takes an essay with a specified segment as input, and generates a sequence of keywords. To improve the correctness and informativeness of generated comments, we modify the keyword sequence by first filtering out incorrect keywords using the error correction module and then inserting structured features from the feature recognition module. Then we feed the polished keyword sequence into the comment generator along with the original essay to get the final comment. During training, we train different modules separately.

3.1 Model Overview

As shown in Figure 2, we propose a planning-based model, which first plans an out-of-order sequence of keywords and then organizes them into a complete comment. Furthermore, we add an error correction module which filters out incorrect keywords using a fine-tuned BERT classifier. We also employ a feature recognition module to recognize fine-grained structured features such as idioms, descriptive and rhetorical methods from the source essay X to enrich the keywords. Moreover, in the comment generation stage, we perturb the input keywords by inserting a random word to alleviate the exposure bias problem. To control the type of the generated comment, we insert a binary control code before generating the keywords and comment.

3.2 Two-staged Planning

Directly generating comments may make models fail to learn specific writing skills and simply over-fit the generic components such as "it is good", which make up the majority of each comment. Therefore, we extract relatively important keywords that have higher TF-IDF (Manning et al., 2010) values than a fixed threshold 0.3 from comments, which are more likely to relate to specific writing skills. Then we employ a comment keyword planner to predict the keywords, and a comment generator to organize them into a complete comment. In order to insert new keywords obtained from the feature recognition module without worrying about insertion positions, we randomly shuffle the extracted keywords. We train the planner and generator by optimizing the negative log-likelihood of ground truths, respectively, formally as follows:

$$\mathcal{L}_{\text{plan}} = -\frac{1}{T}\sum_{t=1}^{T} \log P(k_t|X, k_{<t}), \quad (1)$$

$$\mathcal{L}_{\text{gen}} = -\frac{1}{N}\sum_{t=1}^{N} \log P(y_t|X, K, y_{<t}), \quad (2)$$

where $K = (k_1, k_2, \cdots, k_T)$ with T tokens is the extracted keyword sequence.

3.3 Keywords Filtering and Adding

Intermediate keywords have a significant impact on the quality of generated comments. We observe two main problems in generated keywords: **(1)** The writing skills reflected by some keywords are not used in the source essay (e.g., "echo" in Figure 2), which makes it difficult for the comment generator to generate a correct comment. **(2)** The generated keywords are not enough to cover the used writing skills (e.g., environmental description in Figure 2), which decreases the informativeness of the generated comment. Therefore, we adopt an error correction module and a feature recognition module to modify the keywords and improve the correctness and informativeness of generated comments.

Error Correction Module To filter out incorrect keywords from a keyword sequence, we fine-tune a BERT classifier to predict the probability $P(c_i = 1|X, k_i)$ for a keyword k_i being incorrect, where c_i is the binary label to indicate whether k_i is correct ($c_i = 1$) or not ($c_i = 0$). During training, we take original essay-keyword pairs as positive examples and randomly sampled keywords from

the whole dataset to create the same number of negative examples. We derive the loss function as follows:

$$\mathcal{L}_{\text{cor}} = -\frac{1}{T}\sum_{i=1}^{T}\big(\log P(c_i = 1|X, k_i) + \log P(c_i = 0|X, \hat{k}_i)\big), \quad (3)$$

$$P(c_i|X, k_i) = \text{softmax}(\boldsymbol{W}\mathbf{h} + \boldsymbol{b}), \quad (4)$$

where $\hat{k}_i$ denote the i-th keyword in the randomly sampled keyword sequence, $\mathbf{h}$ denotes the hidden state at the position of the `[CLS]` token and $\boldsymbol{W}$ and $\boldsymbol{b}$ are learnable parameters. The fine-tuned BERT classifier achieves 79.8% accuracy on the test set.

Feature Recognition Module After filtering out incorrect keywords using the error correction module, the keyword sequence may still miss some important features in the source essays. Therefore, we recognize five kinds of fine-grained features from inputs[1].We show several examples of the structured features in Table 1. For idioms, proverbs and quotes, we directly perform word-by-word matching with private off-the-shelf corpora. And we randomly insert the keys of these features into the keyword sequence. We also randomly insert the values of idioms into the keyword sequence. For each kind of descriptive and rhetorical methods[2], we fine-tune BERT as a binary sentence classifier using about 50k manually annotated examples. The fine-tuned BRETs could achieve 92% - 98% accuracy for different kinds of descriptive and rhetorical methods. Then we randomly insert both keys and values of these features recognized by the fine-tuned BERTs into the keyword sequence. We only insert keywords that are not in the sequence to avoid duplication. Finally, we feed the polished keyword sequence into the comment generator to generate the final comment. Note that the comment generator has seen similar fine-grained features extracted by TF-IDF algorithm during training.

3.4 Comment Generation

In the comment generation stage, the lack of exposure to the generated keyword sequence (i.e., the exposure bias issue) may impair the generation performance. To alleviate this issue, we follow Tan et al. (2021) to perturb the input keywords during training. We try various perturbation techniques including replacing a keyword with a randomly sampled one and removing one keyword randomly (Tan et al., 2021), and find that simply inserting a random keyword leads to the best performance in automatic evaluation.

Keys	Values
成语 idiom	无坚不摧 indestructible
俗语 proverb	好事不出门，坏事传千里 bad news travels fast
引用 quote	千里之行，始于足下 a journey of a thousand miles begins with single step
描写方法 descriptive method	动作描写 action description
修辞方法 rhetorical method	比喻 simile

Table 1: Examples for five kinds of fine-grained structured features. Keys indicate the feature type and values indicate the feature content.

4 Experiments

4.1 Dataset

As there is no available dataset for our task, we manually collected a large Chinese dataset to train and evaluate our model. We first collect a large number of pictures of student essays along with comments from professional teachers and the students' grades from an online school[3]. Then we filtered out the essays written by students below the fourth grade to ensure the essays contain abundant writing skills, and retained only narrative essays in this work. Afterwards, we asked crowd-sourced annotators to convert the pictures into texts with the following requirements: **(1)** Correcting misspellings or incorrect punctuation marks; **(2)** Refusing incomplete essays; **(3)** Refusing comments that do not correspond to specific segments; **(4)** Marking the type of comments, i.e., describing strengths or weaknesses. Then we converted marked comment types to binary control codes and insert them before comments and extracted keywords. The detailed statistics are shown in Table 2. We ensure the essays in the training, validation and test sets do not have overlapping titles.

[1] https://openai.100tal.com/documents/article/page

[2] All kinds of descriptive and rhetorical methods are shown in the appendix.

[3] https://www.xueersi.com/

	Train	Valid	Test
# Examples	18,100	2,263	2,036
# Essays	3,996	540	887
Avg. Title Len	5.82	6.06	6.10
Avg. Essay Len	406.89	382.85	345.08
Avg. Number of Par	4.92	4.46	4.28
Avg. Segment Len	97.58	96.15	79.64
Avg. Comment Len	33.92	39.09	38.54
Strength Ratio	84.38%	80.69%	79.47%
Weakness Ratio	15.62%	19.31%	20.53%
Avg. Number of Key	3.65	3.74	4.14

Table 2: Dataset statistics. *Len*/*Par*/*Key* is the abbreviation of *Length*/*Paragraph*/*Keyword*. We compute the length by counting the number of Chinese characters. *Segment* is the specified segment which should be commented on. *Strength/Weakness Ratio* means the proportion of the comments that describe strengths/weaknesses of segments.

Models	B-1	B-2	B-3	B-4	D-3	D-4
GPT2	19.01	11.49	8.90	7.59	7.23	8.65
LongLM	33.40	26.16	22.98	21.13	6.05	7.39
Our Model	**36.16**	**28.32**	**24.87**	**22.86**	9.61	13.56
w/o feature	35.39	27.53	24.14	22.19	9.28	13.05
w/o correct	34.88	26.64	23.07	21.01	11.49	15.73
w/o perturb	33.94	25.58	22.02	19.97	**12.25**	**16.90**
Ground Truth	100	100	100	100	24.81	28.29

Table 3: Automatic evaluation results. The best performance is highlighted in **bold**. All results are multiplied by 100. Note that the components are incrementally removed in the ablation study. For example, *w/o correct* excludes the feature recognition module and the error correction module.

4.2 Baselines

We use LongLM (Guan et al., 2022) as our backbone model, which is pretrained on a large Chinese novel dataset with an encoder-decoder transformer architecture. We also use GPT2 (Radford et al., 2019) as a baseline. They directly generate comments conditioned on the source essays with specified segments. To verify the effectiveness of each proposed component, we exclude them from our model one by one: **(1) w/o feature**: excluding the feature recognition module; **(2) w/o correct**: additionally excluding the error correction module; **(3) w/o perturb**: additionally excluding the perturbations added to the keywords when training the comment generator.

4.3 Experiment Settings

Due to limited resources, we follow $\text{LongLM}_{\text{Base}}$'s hyper-parameters (224M parameters) and utilize the public pretrained checkpoint to initialize our model. We set the learning rate to 3e-5 and batch size to 40. We set GPT2 to the small version with 102M parameters. Other hyper-parameters are the same as LongLM.

In both the planning and comment generation stages, we use top-p sampling with $p = 0.9$ (Holtzman et al., 2020) combined with beam search (number of beams is 4). We only retain comments for automatic evaluation. For the error correction module, we fine-tune a pretrained Chinese BERT (Cui et al., 2020) on auto-constructed data. We set the learning rate to 1e-5 and batch size to 16. For all models, we select the best checkpoint based on the performance on the validation set. To improve the training speed, we train our model on two gpus with mixed precision training and early stop is adopted.

4.4 Automatic Evaluation

Metrics We adopt the following automatic metrics for evaluation on the test set. **(1) BLEU (B-n):** We use $n = 1, 2, 3, 4$ to evaluate n-gram overlap between generated and ground-truth comments (Papineni et al., 2002). **(2) Distinct (D-n):** We use the ratio of distinct n-grams to all the generated n-grams (Li et al., 2016) to measure the generation diversity ($n = 3, 4$).

Result Table 3 shows the automatic evaluation results. Although GPT2 has higher generation diversity, its BLEU score is significantly lower than our backbone model LongLM, suggesting its worse generation quality. Compared with GPT2 and LongLM, our model improves significantly on both BLEU and Distinct scores, indicating higher quality and diversity of the generated comments. As for the ablation study, we can draw the following conclusions: **(1)** Using fine-grained features to enrich the keywords improves both the quality and diversity of the generated comments. **(2)** Error correction module mainly improves the BLEU scores. We note that it has a negative effect on diversity, which suggests the classifier may tend to retain commonly used keywords. **(3)** Adding perturbations to inputs of the comment generator mainly improves the quality of the composed comments as indicated by a higher BLEU score. Besides, through explicitly extracting informative keywords from the comments, we enforce the model to attend on the distinct part of the comments, which greatly improves the generation diversity (comparing *LongLM* and *w/o perturb*). In summary, all

Models	Correctness Win / Lose / Tie	κ	Informativeness Win / Lose / Tie	κ	Coherence Win / Lose / Tie	κ
Ours vs. LongLM	33*/ 12 / 55	0.71	37*/ 15 / 48	0.79	8 / 9 / 83	0.73
Ours vs. Humans	15 / 26 / 59	0.46	24 / 18 / 58	0.77	5 / 11 / 84	0.38

Table 4: Manual evaluation results. The scores indicate the percentage of *win*, *lose* or *tie* (%) when comparing our model with LongLM or humans. κ denotes Fleiss's kappa to measure the inter-annotator agreement. * means the difference is significant with p-value<0.01 (Wilcoxon signed-rank test).

components positively impact the quality or the diversity of the generated comments, and our model strikes a good balance between these two aspects.

4.5 Manual Evaluation

We conduct pair-wise comparisons with LongLM and humans (i.e., ground-truth comments). We randomly sample 100 examples from the test set and obtain 300 examples in total. For each pair of comments along with the input, we hire three well-trained professional annotators to give a preference (win, lose or tie) in terms of three aspects: **(1) Correctness**: whether the strengths or weaknesses identified by the comment are actually present in the segment; **(2) Informativeness**: how much informative information such as "idiom" does the comment contain; **(3) Coherence**, whether the comment is coherent in terms of grammatical correctness, and inter-sentence relatedness, causal and temporal dependencies. Each aspect is evaluated independently and annotators are unaware of the comments source. We adopt majority voting to make the final decisions among three annotators.

As shown in Table 4, our model significantly outperforms LongLM in terms of correctness and informativeness and is comparable with LongLM in coherence. Notably, our model can generate more informative comments than humans thanks to additional information from the feature recognition module, despite the risk of making mistakes. All results show fair ($0.2< \kappa \leqslant 0.4$), moderate ($0.4< \kappa \leqslant 0.6$) or substantial ($0.6< \kappa \leqslant 0.8$) inter-annotator agreement.

We also manually evaluate the controllability of our model to generate two different types of comments (describing strengths or weaknesses). We randomly sample 50 essays from the test set, and generate two comments for each essay to describe strengths and weaknesses, respectively, using different control codes. Then for each example, we ask three well-trained annotators to decide whether the generated comment is consistent with the given control code. We also adopt majority voting to make final decisions among three annotators. We find that 91% of the comments are successfully controlled by the control code and the Fleiss's kappa is 0.85, indicating almost perfect inter-annotator agreement. We conclude that our model has good controllability to generate different kinds of comments so that it can meet the needs of different users for showing praise or giving advice.

Figure 3: An screenshot of our demo website.

5 Demonstration

A screenshot of our demo website is shown in Figure 3. After entering the title and the body of the article, the user can submit a request and get the result after a few seconds. We comment on all paragraphs except those that contain less than 15 Chinese characters and do not have any recognized structured features. Also, we show recognized fine-grained structured features on the right. The sentences and keywords corresponding to these features are underlined and marked green. With these comments and fine-grained features,

the user can fully understand the essay's strengths and weaknesses. Besides the demo website, we also create a github repository at https://github.com/thu-coai/EssayCommentGen, where users can freely use our code and data under the MIT license.

6 Conclusion

We present a planning-based model for a new task named essay comment generation, which first plans a sequence of keywords and then expands these keywords into a complete comment. Furthermore, we utilize an error correction module and a feature recognition module to modify the generated keywords for improving the correctness and informativeness of final comments. We manually collect a new Chinese dataset for this task. Extensive experiments show that our model outperforms strong baselines. We have deployed our model online to help with the automatic essay evaluation. We expect our work to facilitate further research on this new task and benefit both teachers and students.

Acknowledgement

This work was supported by National Key R&D Program of China, under Grant No. 2020AAA0104500. This work was supported by the National Science Foundation for Distinguished Young Scholars (with No. 62125604) and the NSFC projects (Key project with No. 61936010 and regular project with No. 61876096). This work was also supported by the Guoqiang Institute of Tsinghua University, with Grant No. 2019GQG1 and 2020GQG0005, and sponsored by Tsinghua-Toyota Joint Research Fund.

References

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3).

Mădălina Cozma, Andrei 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 (Volume 2: Short Papers)*, pages 503–509.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, 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*, 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.

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660.

Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338.

Jiefu Gong, Xiao Hu, Wei Song, Ruiji Fu, Zhichao Sheng, Bo Zhu, Shijin Wang, and Ting Liu. 2021. Iflyea: A chinese essay assessment system with automated rating, review generation, and recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 240–248.

Jian Guan, Zhuoer Feng, Yamei Chen, Ruilin He, Xiaoxi Mao, Changjie Fan, and Minlie Huang. 2022. Lot: A story-centric benchmark for evaluating chinese long text understanding and generation.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Dongyeop Kang and Eduard H. Hovy. 2020. Self-supervised text planning for paragraph completion task. In *EMNLP*.

Zixuan Ke, Hrishikesh Inamdar, Hui Lin, and Vincent Ng. 2019. Give me more feedback II: Annotating thesis strength and related attributes in student essays. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3994–4004, Florence, Italy. Association for Computational Linguistics.

Xiangzhe Kong, Jialiang Huang, Ziquan Tung, Jian Guan, and Minlie Huang. 2021. Stylized story generation with style-guided planning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2430–2436, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. 2010. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103.

Sandeep Mathias and Pushpak Bhattacharyya. 2018. Thank "goodness"! a way to measure style in student essays. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 35–41, Melbourne, Australia. Association for Computational Linguistics.

Sandeep Mathias and Pushpak Bhattacharyya. 2020. Can neural networks automatically score essay traits? In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–91.

Ellis B Page. 1966. The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. *ICLR*.

Swapna Somasundaran, Michael Flor, Martin Chodorow, Hillary Molloy, Binod Gyawali, and Laura McCulla. 2018. Towards evaluating narrative quality in student writing. *Transactions of the Association for Computational Linguistics*, 6:91–106.

Wei Song, Ziyao Song, Lizhen Liu, and Ruiji Fu. 2020. Hierarchical multi-task learning for organization evaluation of argumentative student essays. In *IJCAI*, pages 3875–3881.

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.

Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric Xing, and Zhiting Hu. 2021. Progressive generation of long text with pretrained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4313–4324.

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 *Thirty-Second AAAI Conference on Artificial Intelligence*.

Chung-Ting Tsai, Jhih-Jie Chen, Ching-Yu Yang, and Jason S Chang. 2020. Lingglewrite: a coaching system for essay writing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 127–133.

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. *Findings of the Association for Computational Linguistics: EMNLP*, 2020:1560–1569.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

A Data Collection

As described in "Section 4.1: Dataset" in our paper, we first collect a large number of pictures of teacher commented student essays and then ask annotators to convert the pictures into texts. We show an example of the original pictures in Figure 4. The corresponding text annotated by humans is shown in Figure 5.

B Manual Evaluation

To perform manual evaluation, we hire well-trained professional annotators from a Chinese crowdsourcing company. For the pairwise comparison evaluation, the annotation instructions are summarized as follows: **(1)** Correctness. Annotators should neglect slight incoherence of the comments and focus on the correctness aspect. If both comments are correct or incorrect, the result should be a tie. Otherwise, the correct comment should be labeled as a win while the other comment should be labeled as a loss. **(2)** Informativeness. Annotators should neglect slight incoherence of the comments and focus on the informativeness aspect. If two comments contain close amounts of informative

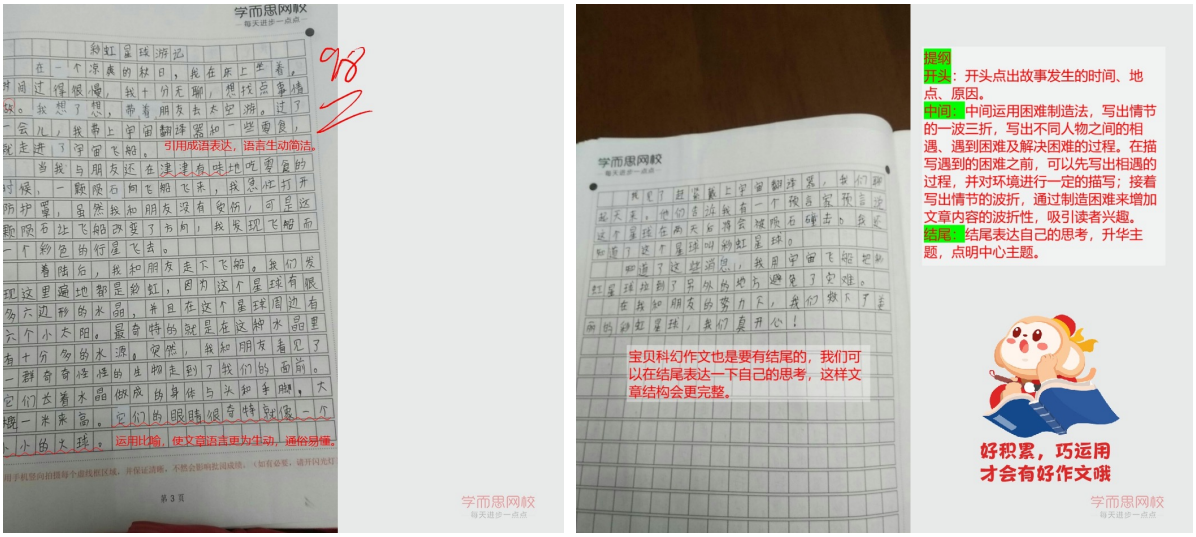

Figure 4: An example of the original pictures of teacher commented student essays.

彩虹星球游记

在一个凉爽的秋日，我在床上坐着。时间过得很慢，我十分无聊，想找点事情做。我想了想，带着朋友去太空游。过了一会儿，我带上宇宙翻译器和一些零食，就走进了宇宙飞船。

当我与朋友还在**津津有味**地吃零食的时候，一颗陨石向飞船飞来，我急忙打开防护罩，虽然我和朋友没有受伤，可是这颗陨石让飞船改变了方向，我发现飞船而是一个彩色的行星飞去。

着陆后，我和朋友走下飞船。我们发现这里遍地都是彩虹，因为这个星球有很多六边形的水晶，并且在这个星球周边有六个小太阳。最奇特的就是在这种水晶里有十分多的水源。突然，我和朋友看见了一群奇奇怪怪的生物走到了我们的面前。它们长着水晶做成的身体与头和手脚，大概一米来高。**它们的眼睛很奇特就像一个小小的火球。**

我见了赶紧戴上宇宙翻译器，我们聊起天来。他们告诉我有一个预言家预言说这个星球在两天后将会被陨石撞击。我还知道了这个星球叫彩虹星球。

知道了这些星球，我用宇宙飞船把彩虹星球拉到了另外的地方避免了灾难。

在我和朋友的努力下，我们救下了美丽的彩虹星球，我们真开心！

片段(segment)	评语(comment)	类型(type)
津津有味	引用成语表达，语言生动简洁。	优点(strength)
它们的眼睛很奇特就像一个小小的火球。	运用比喻，使文章语言更为生动，通俗易懂。	优点(strength)
在我和朋友的努力下，我们救下了美丽的彩虹星球，我们真开心！	宝贝科幻作文也是要有结尾的，我们可以在结尾表达一下自己的思考，这样文章结构会更完整。	缺点(weakness)

Figure 5: An example of the essay with comments annotated by human. The source essay is on the left side and the commented segments are bolded and underlined. The segments along with the comments are shown on the right. Comments could point out the strengths or weaknesses of the segments.

information, the result should be a tie. Otherwise, the more informative comment should be labeled as a win while the other comment should be labeled as a loss. **(3)** Coherence. Annotators should focus on the correctness aspect. If both comments are coherent or incoherent, the result should be a tie. Otherwise, the coherent comment should be labeled as a win while the other comment should be labeled as a loss. We give each annotator ¥1.8 for annotating one pair of comments and one annotator's hourly rate is about ¥108.

For the controllability evaluation, we offer annotators the generated comment and the control token. Annotators should judge whether the comment points out the strengths or weaknesses as the control token specifies. We give each annotator ¥0.5 for annotating one sample and one annotator's hourly rate is about ¥90.

C Keywords Polishing Details

On the test set, we filter out 1.23 keywords using the error correction module and add 1.50 keywords using the feature recognition module for generating each comment on average.

D Structured Features

All descriptive methods include:

- 味觉描写 (taste description)
- 心理描写 (psychology description)
- 嗅觉描写 (smell description)
- 外貌描写 (appearance description)
- 环境描写 (environment description)
- 神态描写 (expression description)
- 语言描写 (language description)
- 动作描写 (action description)
- 视觉描写 (vision description)
- 触觉描写 (touch description)

All rhetorical methods include:

- 比喻 (simile)
- 拟人 (personification)
- 排比 (parallelism)
- 反问 (rhetorical question)
- 设问 (hypophora)