# Integrating Prior Scenario Knowledge for Composition Review Generation

Luyang Zheng[1], Hailan Jiang[2], Jian Wang[1], and Yuqinq Sun[1(✉)]

[1] School of Software, Shandong University, Jinan, China
sun_yuqing@sdu.edu.cn
[2] Shandong Polytechnic, Jinan, China
1792@sdp.edu.cn

**Abstract.** Reviews generation is an important task for the elementary and middle school composition evaluation. Existing methods only focus on reviewing the composition contents without consideration of the effects of different grades and the types of the composition. To solve this problem, we proposed a light-weight composition review generation method. It incorporates the scenario dependent prior knowledge to reflect the diverse requirements of composition evaluation across different grades and writing styles, which includes three parts, the frequent word bank, the prior token distribution and the domain prior distribution. To incorporate scenario-dependent prior knowledge, we first encode the composition and the frequent word bank. Then we use the cross-attention mechanism to integrate the frequent word bank, and use the dynamic weight to integrate the prior token distribution into the decoding state, separately. Meanwhile, the correction module is designed to calibrate the style of the generated reviews based on the domain prior distribution. Our proposed method is compared with the SOTA works on real datasets. The experimental results demonstrate that it outperforms several strong baselines and the reviews generated by our method are better than those generated by the baselines in terms of fluency, correctness and rationality. Additionally, to verify our method's potential applicability to other review tasks, we transfer our method to the task of generating reviews for academic papers and the results show the effectiveness of our method.

**Keywords:** prior knowledge · composition review · text generation · scenario · dynamic fusion

## 1 Introduction

Composition can comprehensively reflect the students' writing ability and knowledge mastery level, which is a common form of inspection in the primary and secondary education. To help students better recognize the strengths and weaknesses in their compositions, it is necessary not only to give scores but also to generate composition reviews. There are different grade specific requirements on composition evaluation. For example, for third-grade compositions, reviewers focus more on the completeness of the composition, whereas for ninth-grade compositions, more focus should be on the intent of compositions. Besides, different


C. Cao et al. (Eds.): KSEM 2024, LNAI 14884, pp. 298–309, 2024.
https://doi.org/10.1007/978-981-97-5492-2_23

types of compositions are often associated style-specific review requirements. In conclusion, in the primary and secondary school composition evaluation, besides the composition content, scenario constraints such as the student's grade and type of composition can affect the review. The aim in this paper is to generate the composition reviews that satisfy scenario constraints. Also, some specific composition evaluation scenarios are not allowed to connect to the Internet and offer limited cost for localized deployment. So we should provide lightweight models for these scenarios.

Existing composition review generation methods [5,22] typically design multiple analysis modules to understand the composition and generate reviews based on the analysis results. However, these methods generate reviews based only on the composition content, and analysis modules are designed for general scenarios. So the reviews cannot reflect the scenario constraints. Moreover, the controlled text generation methods [1,2,10,17] are also relevant to our problem. However, because the attributes introduced in these methods are general, such as emotion and theme, these methods applied to our problem cannot reflect the differences in composition evaluation criteria for different scenarios. Currently, although the large language models have shown excellent performance on many text generation tasks, it cannot be directly applied in some scenarios without Internet. Fine-tuning such models specifically for these scenarios requires large computational power and labeled data, which cannot be satisfied in real-world scenarios. Therefore, the large language models cannot be applied to our task either.

To solve the above problems, we propose a composition review Generation model using Prior scenario Knowledge, PKG for short. To mine the information about scenario constraints, we constructed three types of prior scenario knowledge, namely the frequent word bank, the prior token distribution, and the domain prior distribution. The first two types reveals the effect of different scenario constraints on reviews at the lexical, token level, while the last one indicates the overall style and characteristics of the review in the specific assessment domain. We first encode the composition content and the frequent word bank separately by the text and knowledge encoder. In the generation process, we designed three modules. The frequent word space fusion module uses the cross-attention mechanism to merge the frequent word bank. To control the generation more directly, the prior distribution dynamic fusion module combines the predicted vocabulary distribution with the prior token distribution by dynamic weight. To make the generated reviews more similar to the style of the real reviews, we designed the correction module. We introduce the vocabulary feature map to represent the overall vocabulary distribution for the generated text. Then, we used the correction loss to minimize the difference between the domain prior distribution and the vocabulary feature map. This way of correction is for the vocabulary distribution, not for the specific words, which benefits the fast convergence of our model. Empirical results demonstrated that PKG outperforms strong baseline models and the generated reviews by PKG are more accurate, fluent and can satisfy the scenario constraints.

## 2 Related Work

Existing methods for review generation mainly use the source text and the near-neighbor information about the source text. To mine deeper information from the source text, Li et al. [9] organized the news as a topic interaction graph. It helps the model better understand the structure of news and the connection between topics. Moreover, some researchers design different ways to mine the source text and introduced different forms of external knowledge. Gong et al. [5] present a Chinese assessment system. They design multi-level analytical modules and combine the analysis results with pre-defined templates to generate reviews. Due to the lack of flexibility of templates, Zhang et al. [22] proposed a planning-based model. It plans some keywords related to a specific writing skill, then predicts the review keywords based on the composition, and finally expands these keywords into a coherent review through a language model. Yuan et al. [19] proposed an knowledge-guided model for academic paper reviews, which incorporates the knowledge: citation graph and concept knowledge from the content.

Furthermore, controlled text generation methods are also related to our task, which often leverage the pre-trained language model(PLM) as the backbone network. The first type of approach is to guide the model through attribute representation. Some studies try to introduce control modules during model training stage to fine-tune PLM [1] , while others directly associate attributes with specific embedding representations for the controllable text generation [18]. The second type is to fine-tune the PLM by introducing attribute-dependent near-neighbor information. The near-neighbor information can be either graph-structured information [19] or textual information [7,17]. The third type of approaches is to fine-tune the PLM based on model feedback information, like discriminators, reinforcement learning, where the most representative methods are PPLM [2]. The fourth type generates controllable text by prompts, e.g., using trainable prefixes [21] or natural language text as prompts [10].

To sum up, the above methods are more focused on exploring the connection between the source text and the target text in various ways to better guide the decoding process.

## 3 Methodology

### 3.1 Review Generation Incorporating Prior Scenario Knowledge

By analyzing various real composition evaluation standards, we found that the composition reviews implied evaluation criteria for different grades and composition types. For each scenario based on students' grade and composition type, there are some common traits in the reviews that reflect the specific criteria and evaluation perspectives. For example, the words "rhetoric devices" and "vivid" often appear in the reviews of narrative composition, while the word "abundant evidence" often appears in the reviews of argumentative composition. Therefore, we construct the prior scenario knowledge to reflect these evaluation criteria.

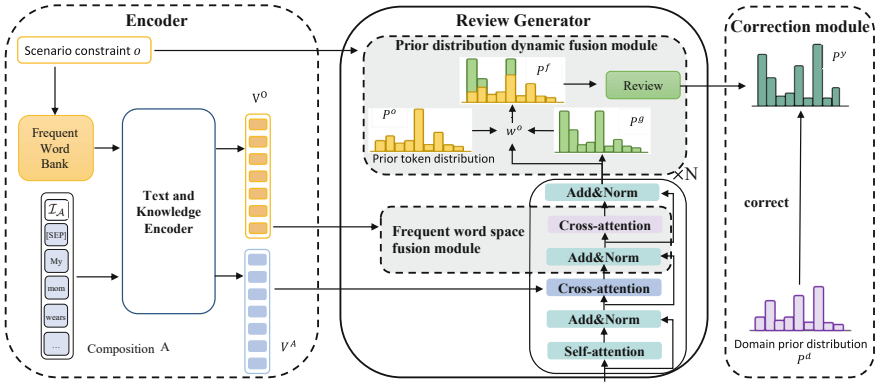



**Fig. 1.** Overview of our PKG model.

Firstly, we take the tuple of grade and type of each composition as its scenario constraint $o$, which are the two important factors. For each kind of scenario constraint, we construct three kinds of prior knowledge: 1)**frequent word bank** $S_o$: We treat the top-$m$ frequent words of the reviews on $o$ as $S_o$, i.e. $S_o = \{w_1, w_2, \ldots, w_m\}$ and $w_i$ is the $i$-th frequent word; 2)**prior token distribution** $P^o$: For each kind of $o$, we get the $P^o$ by counting the token frequency of all reviews belong to $o$ and dividing these frequencies by reviews' number;3)**domain prior distribution** $P^d$: $P^d$ indicates the overall style of the reviews. We count the frequency of all tokens on the vocabulary and divide these frequencies by the total number of tokens in all reviews of the dataset.

Our proposed model consists of three parts, which are the text and knowledge encoder, the review generator and the correction module, shown as Fig. 1. Firstly, to incorporate the scenario-related prior knowledge, we encode the composition and frequent word bank separately by our encoder. In the decoding process, we design two modules in the generator, namely the frequent word space fusion module and the prior distribution dynamic fusion module. Lastly, we design the correction module to check the style of the generated reviews.

### 3.2 Encoder and Review Generator

**Text and Knowledge Encoder.** To make our model understand the composition more directly, we extract the top-$k$ important sentences of the composition $x$ as its abstract $A$ by the unsupervised TextRank algorithm [13]. It puts all composition lengths within the range of inputs that PLM can handle.

We add different prompts in the form of text for both $A$ and the frequent word bank $S_o$ to help the encoder recognize different inputs. We denote the prompt of $A$ as $\mathcal{I}_\mathcal{A}$. Then we use the $[SEP]$ token to concatenate $\mathcal{I}_\mathcal{A}$ and $A$, and feed it into the encoder to get the semantic representation of $A$ as $V^A = Encoder(\mathcal{I}_\mathcal{A}; [SEP]; A)$. To encode the $S_o$, we firstly concatenate all the words contained in $S_o$. However, since the words are discrete, we add the $[C]$ token between two adjacent words to help the encoder understand each word. After the above steps, we get the frequent word information

$O = \{w_1; [C]; w_2, [C]...; [C]; w_m\}$. Then we concatenate the prompt of the frequent word information $\mathcal{I}_{\mathcal{O}}$ and $O$ as above and input it into the encoder to get the representation of $O$ as $V^O = Encoder(\mathcal{I}_{\mathcal{O}}; [SEP]; O)$.

**Frequent Word Space Fusion Module.** To make the generator focus on the lexical information of the $V^O$, we use the cross-attention mechanism. We add the frequent word space fusion module to each decoder layer, as shown in Eq. (3). By this module, $V^O$ is integrated into the generator with the attention of the current decoding state to $V^O$. We denote $h_l$ as the representation of output in $l$-th decoder layer.The $(l+1)$-th decoder layer output $h_{l+1}$ is obtained as follows:

$$h_{l+1} = LN(h_l + SelfA(h_l)), \tag{1}$$

$$h_{l+1} = LN(h_{l+1} + CrossA(h_{l+1}, V^A)), \tag{2}$$

$$h_{l+1} = LN(h_{l+1} + CrossA(h_{l+1}, V^O)), \tag{3}$$

$$h_{l+1} = LN(h_{l+1} + FFN(h_{l+1})), \tag{4}$$

where $LN(\cdot)$, $FFN(\cdot)$, $SelfA(\cdot)$ and $CrossA(\cdot)$ represent layer normalization, feed-forward neural network, self-attention and cross-attention module respectively. There are a total of $L$ decoder layers, and the output of the last layer is $h_L$. We input it into a fully connected layer and a softmax layer to obtain the vocabulary distribution $P^g \in R^v$, where $v$ is the size of the vocabulary.

**Prior Distribution Dynamic Fusion Module.** Since the model generates the reviews based on the final predicted vocabulary distribution, changing it controls the generated reviews more directly. Considering the similarity between the prior token distribution $P^o$ and the $P^g$, we designed the prior distribution dynamic fusion module to combine distributions by dynamic weights.

We feed the $h_L$ into a multi-layer perception and a softmax layer to obtain the dynamic weights $w^o \in R^v$. And we update the $P^o$ according to the formula $P^o = w^o \bigodot P^o$, where $\bigodot$ represents the element-wise multiplication. To obtain the final output distribution $P^f$, we combine the $P^o$ and the $P^g$ as $P^f = \frac{P^o + P^g}{2}$. The loss function of generation is obtained as $loss_g = \sum_{i=1}^{T} -\log P^f(y_t|x, y_{<t})$, where $y_t$ is the $t$-th token in the real review, $T$ is the length of the generated review, $y_{<t}$ is reviews generated by the first $t$ step, $P^f(y_t|x, y_{<t})$ is the output probability at time step $t$ of the generator.

### 3.3 Domain Prior Distribution Guided Review Correction

To make the generated reviews more realistic, we design the correction module to use the domain prior distribution for correction. In the training stage, to avoid the gradient not being derivable, we first obtain the vocabulary feature map $P^y$ for the generated reviews $y$. For each sample, $P^y$ is calculated by averaging the final output vocabulary distributions at each time step as $P^y = \frac{1}{T}\sum_{i=1}^{T} P_i^f . P_i^f$ is the final output vocabulary distribution at time step $i$. Because the domain prior distribution $P^d$ is obtained from the all reviews in the dataset, the $P^y$ should be similar to it. So we design the correction loss function $loss_c$ to minimize the

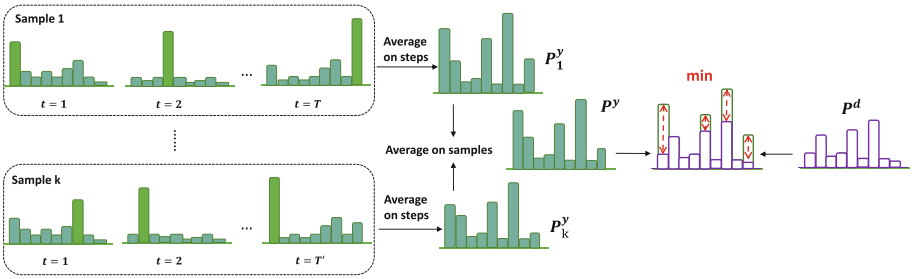



**Fig. 2.** The correction loss based on a batch of samples.

difference between $P^y$ and $P^d$, which is calculated as $loss_c = \frac{\|P^y - P^d\|_1}{Max}$, where $Max$ denotes the maximum value of the absolute value of all elements in the $P^y - P^d$, $\|\cdot\|_1$ is the sum of each element. To scale the value of $loss_c$ and make its value in the magnitude of $loss_g$, the division by $Max$ is performed. It also makes model optimization easier. Since $P^y$ and $P^d$ are not real probability distributions, we cannot use the common method of measuring distribution divergence, such as KL divergence.

To increase the diversity of reviews, we modify the $P^y$ to be the average of all samples' vocabulary feature maps in a batch when computing $loss_c$. For example, we set $batch\_size = k$ as shown in Fig. 2 and modify the $P^y$ using the formula $P^y = \frac{1}{k}\sum_{j=1}^{k} P_j^y$, where $P_j^y$ is the vocabulary feature map of the generated review of the $j$-th sample. The correction module is used only in the training stage. In the inference stage, the review is directly obtained from $P^f$.

At last, we optimize the PKG model uniformly by combining $loss_g$ with $loss_c$. The overall loss function $\mathcal{L}$ is calculated as formula $\mathcal{L} = \alpha loss_g + (1-\alpha)loss_c$, where $\alpha$ is a hyper-parameter.

## 4 Experiments

### 4.1 Datasets and Baselines

Experiments are conducted on a Chinese **Composition Review(CR)** dataset. Subsequently, we also conducted experiments on the **ASAP-Review(AR)** dataset of academic papers to validate the migration of our PKG method. The statistical information of these datasets is shown in Table 1. Total is the overall sample size. Con, Sum, and Rev are the average number of tokens of the source text, the summary, and the review respectively. The **CR** dataset contains compositions for students from grades 4 to 12. Each composition is labeled with the title, content, review, grade, type, and the score. Since some of the reviews were short and contained little valid information, we filtered the sample for reviews less than 50 tokens in length. The **AR** dataset [19] is for the task of academic paper review generation. It contains submitted papers from NeurIPS 2016–2019 and ICLR 2017–2020. Each sample in this dataset contains the title, introduction, abstract, review, conference, and whether the paper was accepted or not.

**Table 1.** Statistics of the datasets.

| | CR dataset | | | AR dataset | | |
|---|---|---|---|---|---|---|
| | Train | Dev | Test | Train | Dev | Test |
| Total | 11578 | 1450 | 1450 | 10231 | 1271 | 431 |
| Con | 752.7 | 755.4 | 750.4 | 733.11 | 736.38 | 752.83 |
| Sum | 422.4 | 422.6 | 422.4 | 168.50 | 169.09 | 166.4 |
| Rev | 92.7 | 92.1 | 93.4 | 389.52 | 386.93 | 386.97 |

The baseline models we used are as follows: 1)**BPGN**: Based on the pointer generator model [16], we combined a pre-trained BERT model [4] and a layer of LSTM [6] as its new encoder; 2)**BART** [8]; 3)**T5** [15]; 4)**KID** [19]: KID generates knowledge-guided paper comments based on the BART model by introducing citation graphs and concept graphs. For automatic evaluation, we used ROUGE [11] and BLEU [14] to analyze the overlap of ngrams between the generated reviews and real reviews. We evaluate the fluency and correctness of the generated reviews using METEOR [3]. It calculates the harmonic mean of the accuracy and recall between the generated and the real by introducing the chunks. Also, we used BARTScore [20] to assess the coherence of the generated reviews.

### 4.2 Experimental Settings

PKG adopts the BART-base as the backbone. On the CR dataset, we concatenate the composition title and the abstract as the input. We count the frequent word bank of all reviews on different grades and types and set $m$ to 20. On the AR dataset, the title and introduction are concatenated as the input. The tuple of the conference and acceptance is used as the scenario constraint. We count the frequent word bank on different acceptances and conferences and set $m$ to 40. In the experiments, to allow our model to converge quickly, the divisor was changed to the number of reviews when calculating $P^d$, and the length of reviews was not divided when calculating $P^y$. In the training stage, we set the $\alpha$ to 0.6 for the CR dataset and 0.7 for the AR dataset. We use AdamW [12] as the optimizer. In the decoding stage, both PKG and the baseline models use beam search and the beam size is set to 4. The learning rate of PKG is set to 4E-5 for the CR dataset. For the AR dataset,the learning rate of the BART model part in PKG is set to 4E-6, and other parts is set to 0.001.

### 4.3 Model Performance Comparison

We compared the performance of the different models on the CR dataset, as shown in the first block of Table 2. It shows PKG obtains higher scores than baselines in all metrics. Observing ROUGE and BLEU, PKG has significantly improved the accuracy of ngrams compared with other models. The reason is PKG can better understand the compositions and incorporate the prior knowledge into the generation. Compared with BART, ROUGE-1/2/L of PKG are 1.5, 1.8 and 1.77 points higher respectively. It reveals the three modules learn the

**Table 2.** Model comparison.

| Dataset | Models | Rouge-1 | Rouge-2 | Rouge-L | METEOR | BLEU-3 | BLEU-4 | BARTScore |
|---|---|---|---|---|---|---|---|---|
| CR | BPGN | 25.90 | 7.00 | 23.46 | 27.03 | 2.29 | 0.93 | −4.09 |
| | T5 | 28.90 | 6.63 | 21.10 | 25.54 | 2.07 | 1.11 | −4.23 |
| | BART | 36.79 | 11.83 | 32.35 | 33.77 | 3.05 | 2.17 | −3.88 |
| | MRG | **38.38** | **13.63** | **34.12** | **34.76** | **3.79** | **2.84** | **-3.80** |
| AR | BPGN | 15.49 | 1.69 | 10.23 | 8.81 | 0.01 | 0.001 | −5.75 |
| | BART | 26.39 | 5.26 | 12.59 | 16.49 | 0.84 | 0.24 | −5.19 |
| | KID | 13.53 | 4.25 | 6.95 | 8.48 | 0.04 | 0.01 | −5.58 |
| | PKG | **27.88** | **7.21** | **13.65** | **18.51** | **1.18** | **0.29** | **−5.12** |

**Table 3.** Ablation experiments on PKG.

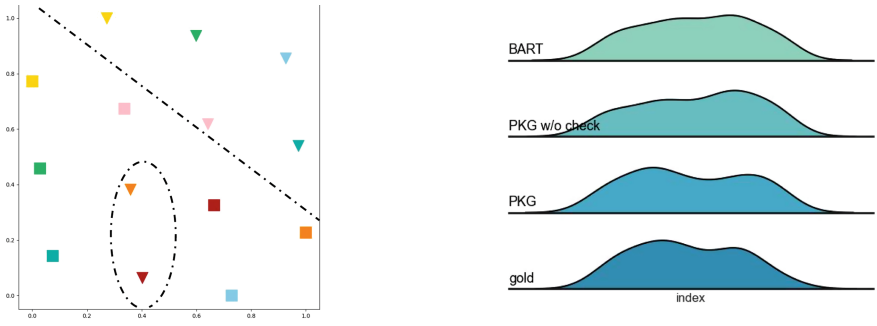
| Models | Rouge-1 | Rouge-2 | Rouge-L |
|---|---|---|---|
| PKG | 38.38 | 13.63 | 34.12 |
| w/o dis-merge | 37.91 | 13.39 | 33.91 |
| w/o check | 37.76 | 13.19 | 34.11 |
| w/o frequent words | 37.37 | 12.41 | 32.92 |

lexical-level and the token-level characteristics of the real reviews, which make the generated reviews more realistic. The results of METEOR and BARTScore show the reviews of PKG are more fluent than others. In addition, the real review is subjective and the evaluation may not be comprehensive. If we only use the real review for training, it will make the generated results heavily depend on the real review and lose diversity. That is why we introduce prior scenario knowledge and the results of all evaluation metrics verify the validity of our method.

To verify that our PKG model can be adapted to different domains, we select the task of review generation for English academic papers. The second block of Table 2 shows the results on the AR dataset. It shows PKG obtains higher scores than baselines in all metrics. Compared with BART, ROUGE-1/2/L of PKG are 1.49, 1.95 and 1.06 points higher respectively. The results of METEOR and BARTScore also show the PKG'S reviews are more fluent. It suggests that our method can be adapted to other domains. Additionally, the performance of the KID is low, which may be caused by the insufficient length of the generated text. But PKG can capture as much information as possible in a limited length, which also confirms the validity of the PKG method from another perspective.

### 4.4 Model Ablation Experiments

To show the validity of each module in PKG, we conduct the ablation experiment on the CR dataset. It removes the following parts from the model respectively: (1)w/o dis-merge: remove the prior distribution dynamic fusion module in the training stage and predict the output based only on the generated vocabulary distribution $P^f$. (2)w/o check: remove the correction module in the training stage. (3) w/o frequent words: remove the frequent word space fusion module.




(a) (b)

**Fig. 3.** Visualization results of the prior token distributions and model outputs.

The results are shown in Table 3. We can find that after removing the frequent word space fusion module(w/o frequent words), the performance drops the most. It shows that if the constraint is converted into fine-grained information, our model can learn better. Meanwhile, this lexical-level prior knowledge can help the model construct a more reasonable review space so that the model can generate desired reviews. After removing the correction module(w/o check), ROUGE-1/2 has decreased, while ROUGE-L has hardly changed. It shows that this module improves accuracy without sacrificing fluency.

Then, we explore the differences among the prior token distributions of each scenario constraint on the CR dataset. Only if there are differences in these distributions can the PKG model learn how to flexibly generate reasonable reviews based on different scenario constraints. We reduce the dimensionality of each prior token distribution and the visualization results are shown in Fig. 3a. The dot represents the prior token distribution of each scenario constraint. The triangles and squares represent narrative and argumentative compositions, respectively. Different colors indicate different grades. We find the dashed lines can divide dots by shape, which shows there are similarities between prior token distributions of the same composition type. And the dots are scattered. It means the reviews of different scenario constraints are diverse in tokens, and the prior token distributions can reflect these token-level differences. Therefore, it is because of the differences between the prior token distributions that our PKG method is able to learn the features of the reviews for different scenarios.

To verify that the domain prior distribution serves as correction, we also design a visualization experiments. We counted the token frequency in each of these four reviews, i.e. the real review and the reviews generated by BART, PKG after removing the correction module (PKG w/o check) and PKG. The results are shown in Fig. 3b. The horizontal axis denotes the token index and the vertical axis denotes the frequency of each token. Comparing the PKG w/o check with BART, we can find that the frequency word bank and prior distribution will cause the token distribution to change to a certain extent, which also achieves the purpose of controlling the generation of reviews. Meanwhile, by comparing

**Table 4.** Case Study on the reviews by PKG and baselines on the CR dataset.

| **Scenario constraint**: Grade 4, Narrative composition describing events |
|---|
| **Gold:**这是描写一篇叙述观灯的记叙文,文章脉络清晰可见,对“后羿射日”、“愚公移山”、“神舟四号”和“三峡大坝”的描写生动具体,语言活泼明快,富有情趣,大量采用生活中的口语,使文章更显自然亲切。It is a narrative composition about lantern viewing. It has a clear structure with vivid and specific descriptions of "Hou Yi Shoots the Sun", "Yu Gong Moves Away the Mountains", "Shenzhou IV" and "Three Gorges Dam". The language is lively and bright, full of interest. It uses lots of spoken language, making the article more natural and friendly. |
| **BART:**这是一篇写景的作文,语言活泼、轻快,“到处可以看到非常醒目的八个大字:‘尧庙灯会,中华之最”、“神舟四号”,“三峡大坝”等比喻和拟人的修辞手法将灯展描写得十分美丽和充满生机,行文流畅,结构安排合理,前后照应,结尾直抒胸臆,表达了作者对古代人民的自豪之情,立意浅显而耐人寻味,富有丰富的意蕴,引起读者的阅读兴趣,结尾点题,表达了作者对古代人民的喜爱之情,升华了主题,给人启发。希望小作者继续努力。It is *a composition about sights*. The language is lively and light. *The rhetorical devices of metaphor and personification*, such as "we could see the big letters everywhere: Yao Temple Lantern Festival, the best in China", "Shenzhou IV" and "Three Gorges Dam", depict the light show as beautiful and full of vitality. The writing is smooth and the structure is reasonable. The author expresses feelings directly at last and the pride of the ancient people. The theme is simple and intriguing, rich in meaning, arousing readers' interest. The ending voices the author's love for the ancient people, sublimating the theme and inspiring us. I hope the young author continues to work hard! |
| **PKG:**这是一篇叙事的文章,小作者详细地叙述了观看尧庙灯展的过程。文章语言自然流畅,用词准确生动,特别是词语“灯火辉煌”、“富丽堂皇”、“繁华似锦”用得好。文章结尾处点明了主题,表达了作者对古代人民的自豪之情。如:“古代人民是多么的伟大,他们在这片神秘的土地上,创造了神话般的东方文化、东方文明”等。文章采用总分总形式,字里行间流露出作者对古代人民的敬佩之情。全文结构完整,结尾处作者直抒胸臆...It is *a narrative composition*, the young author *narrates the process of watching the Yao Temple's Lantern in detail*. The language is natural and fluent. The words are accurate and vivid, especially the words "ablaze with lights" and "gorgeous" which are well used. The theme is pointed out at last, which expresses the author's pride for the ancient people, such as "how great the ancient people were, they created the mythical Eastern culture and Eastern civilization in this mysterious land" and so on. The article adopts the form of deduction and summary, and the author's admiration for the ancient people is revealed. The structure is complete, and the author speaks his mind at the end... |

PKG w/o check, PKG, and gold, we can find that the correction module check the review to match the real distribution better.

### 4.5 Case Study

To show the quality of the generated reviews, we present the reviews generated by different models for the following example.

*Example 1.* This is a fourth-grade narrative composition titled Lantern Viewing. The content of the composition is as follows: *On the evening of February 7, my*

*parents took me to watch the Yao Temple Lantern Festival. We took a cab and drove to Yao Temple...(omitted some content)On the way back, I kept thinking: how great the ancient people were, they created the mythical Eastern culture and civilization in this mysterious land, and I feel proud to live in this land!*

Table 4 shows the generated reviews of all models. To identify the key information in reviews, we marked Chinese with different color and English with italics. The review generated by BART defines the wrong type of composition and describes the details incorrectly(marked in italics). The shown case is a narrative article, but BART defines it as an article about sights, and the rhetorical devices mentioned by BART are not used in the example. PKG can not only accurately define the composition type, but also correctly summarize the composition (marked in italics). In addition, compared with BART, the result of PKG also includes multiple aspects, such as theme, expression, and structure. It shows that our three modules can guide the PKG model to evaluate the composition from various aspects without deviating from the content of the real review.

## 5 Conclusion

In this paper, we propose a composition review generation method incorporating the prior scenario knowledge. We provide a strategy for constructing prior scenario knowledge. We first encode the composition content and the frequent word bank. And then, to generate reviews conform to the scenario constraint, we design three modules to integrate the prior knowledge into the generation process in a reasonable way. We conducted experiments on the real review datasets. The experimental results demonstrate our method can generate more fluent, reasonable reviews and can be well adapted to the review generation tasks in other domains. In the future, we would like to further explore how to generate high-quality objective composition reviews in few-shot learning that do not rely only on the reference reviews.

**Acknowledgments.** This work was supported by the National Natural Science Foundation of China (62376138) and the Innovative Development Joint Fund Key Projects of Shandong NSF (ZR2022LZH007).

## References

1. Chan, A., et al.: CoCon: a self-supervised approach for controlled text generation. In: 9th International Conference on Learning Representations (2021)
2. Dathathri, S., et al.: Plug and play language models: a simple approach to controlled text generation. In: 8th International Conference on Learning Representations (2020)
3. Denkowski, M.J., Lavie, A.: Meteor universal: language specific translation evaluation for any target language. In: Proceedings of the Ninth Workshop on Statistical Machine Translation, pp. 376–380 (2014)
4. Devlin, J., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics, pp. 4171–4186 (2019)

5. Gong, J., et al.: 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, pp. 240–248 (2021)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Hu, Z., et al.: PLANET: dynamic content planning in autoregressive transformers for long-form text generation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, pp. 2288–2305 (2022)
8. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880 (2020)
9. Li, W., et al.: Coherent comments generation for Chinese articles with a graph-to-sequence model. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, pp. 4843–4852 (2019)
10. Li, X., et al.: Unified demonstration retriever for in-context learning. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, pp. 4644–4668 (2023)
11. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)
12. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: 7th International Conference on Learning Representations (2019)
13. Mihalcea, R., Tarau, P.: Textrank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004)
14. Papineni, K., et al.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
15. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020)
16. See, A., Liu, P.J., Manning, C.D.: Get to the point: summarization with pointer-generator networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp. 1073–1083 (2017)
17. Xie, Z., et al.: Factual and informative review generation for explainable recommendation. In: Thirty-Seventh AAAI Conference on Artificial Intelligence, pp. 13816–13824 (2023)
18. Yu, D., Yu, Z., Sagae, K.: Attribute alignment: controlling text generation from pre-trained language models. In: Findings of the Association for Computational Linguistics, pp. 2251–2268 (2021)
19. Yuan, W., Liu, P.: Kid-review: knowledge-guided scientific review generation with oracle pre-training. In: Thirty-Sixth AAAI Conference on Artificial Intelligence, pp. 11639–11647 (2022)
20. Yuan, W., Neubig, G., Liu, P.: Bartscore: evaluating generated text as text generation. In: Annual Conference on Neural Information Processing Systems, pp. 27263–27277 (2021)
21. Zhang, H., Song, D.: Discup: discriminator cooperative unlikelihood prompt-tuning for controllable text generation. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 3392–3406 (2022)
22. Zhang, Z., et al.: Automatic comment generation for Chinese student narrative essays. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 214–223 (2022)