KASG: Korean Author-specific Story Generation

Korea University COSE461 Final Project

Yein Park
Department of Linguistics
Team 14
2017131521

Yoonjin Oh
Department of Business Administration
Team 14
2020120036

Abstract

Lots of researches have been conducted throughout story generation domain with LLM. In story generation research field, consistent and coherent text generation is important, but difficult to control. To address this, we present Author-specific Story Generation model fine-tuned with LoRA technique, to create a text concerning author's original style. Typically, we focus on the Korean epic novel, *Land* written by *Park, Kyung-Ree* then experiment with 4 cases of model that can mimic the author's writing style. The results of our evaluation demonstrate that the proposed method was quite effective, though still hard to get a readable texts.

1 Introduction

With recent advances in natural language generation achieved by the transformer architecture and large scale pre-trained language models, more researchers have leveraged natural language processing techniques to story generation. In general, natural language generation is performed based on stochastic prediction. Specifically, the pre-trained neural models learn large text corpora with log-likelihood objectives by predicting the next token based on previous context. This objective fits well with the characteristics of natural language in which content in written in one direction. So the results generated by the language model, fine-tuned using the story corpus, were quite natural, just like a sentence written by an actual human. However, in story generation, it has distinct limitations. Because language models generate text based on the probability distribution predicted from a given sentence, they cannot control the overall content, theme, or style of a story when generating text.

As mentioned above, Generating a consistent and coherent story is a long-standing task in story generation with NLP. Maintaining a consistent flow of story and characters is important to prevent readers from losing their attention. Therefore, controlling those attributes is a key to the success of an automated story generation. Recent success of story generation enables language models to generate text condition on a specific control code such as sentiment, topic, genre and so on. Inspired by these attempts, we propose author's writing style as a key attribute for automated story generation. The author is a creator of the novel. As the work contains his ideas, experiences, perspectives, and values, the author's impact on the work is considerable. Specifically, the novel's overall structure including main event, character, setting and topic is reflected by the author's interests through the novels. In most cases, each author has his own writing style or interest. Moreover, some authors have their favored topic or messages to deliver. In this aspect, we try to use author as a key attribute for generation and make a sample model in this context, which can be developed into the purpose for mimicking deceased author, or helping to end unfinished works.

2 Related Work

In this section, we discuss related works with respect to the controllablility in story generation, and LoRA mechanism.

2.1 Controllability in Story Generation

In story generation, previous studies obtain Controllability by attempting to condition on story structure (e.g. event) or attribute (e.g. topic, sentiment). In Tambwekar et al.[08], the authors formalized story generation as a planning problem. The authors proposed a reinforcement learning method using the RNN-based language model to create stories toward a given goal. Fanet al.[09] proposed a story generation system that creates a story structure considering action plan and entities before generating the full story. Peng et al.[10] uses the ending valence as the control factor. The model receives the same storyline but a different ending valence(eg, happy or sad); it creates a story with the desired ending. In contrast, in Fan et al.[11], the description of the overall topic of the story is used as a control factor. The authors propose a hierarchical framework that generates the description first and then creates story texts. Yao et al.[12] controlled a text with a title for open-domain story generation. Zhexin Zhang et al.[05] controlled a protagonist's persona in story generation. The authors propose a planning-based generation model named CONPER to explicitly model the relationship between personas and events. Our study differs from these studies in that we control the overall storyline by author, which is a suitable control factor for story generation as described in section 1.

2.2 LoRA(Low Rank Adaptation of Large Language Models)

The increasing size of language models raises great research interests in parameter-efficient fine-tuning such as LoRA. Edward Hu et al.[03] first proposed LoRA: Low-Rank Adaptation of Large Language Models. It freezes the pre-trained mode, and injects small-scale trainable parameters for multiple downstream tasks. Studies about text generation using LoRA as a fine-tuning technique is as follow. Yunqi Zhu et al.[06] proposed a framework that integrates LoRA and structured layer prunning to create deindentified medical report summarization.

3 Approach

In this section, we propose story generation model based on pre-trained SKT KoGPT2 with a methodology called LoRA(Low Rank Adaptation of Large Language models). We first load the pre-trained KoGPT2 API for tokenizer and model, then fine tune it using the novel text of Korean writer, *Park, Kyung-Ree*.

3.1 Conditional Language Model

LM(Language Model) uses probability distribution $p(x)$ for the variable length of text sequence like $x = x_1, x_2, ...x_n$. For each token x_t came from vocabulary set, distribution is set by the equation below.

$$P(x) = \prod_{t=1}^{|x|} P(x_t|x_{<t}) \tag{1}$$

The equation decodes a continuation of x using predictions of model. It uses trainable parameters of the LM, as the LM can learn $p_\theta(x_t|x_{<t})$. Then, the continuation is generated repeatedly with sampling a next token from the previous token of sequence. So, it finds θ and minimize the negative log likelihood loss. The equation of Loss is like below.

$$L = -\sum_{i=1}^{|D|} \sum_{t=1}^{|x(i)|} logP(x_t^i|x_{<t}^i) \tag{2}$$

3.2 Model Architecture

The conditional LM we used for this project, is based on the transformer architecture. It is KoGPT2 by SKT and it has 125M parameters. Basically, it consists with a multiple transformer decoder blocks with masked multi head self attention, layer normalization, and position-wise feed forward operation.

3.3 Code Basis

We referenced a code basis of 'KoGPT2novel' for fine tuning KoGPT2 with a novel text [01]. The learning mechanism uses *fastai* library and it only relies on the CPU environment for fine tuning the whole model. We changed it into a GPU environment and did the experiment by 'Colab Pro +' which provides A100 GPU and high dose of RAM for its run-time.

3.4 LoRA(Low Rank Adaptation of Large Language Models)

'LoRA(Low Rank Adaptation of Large Language Models)' is one of techniques of PEFT(Parameter Efficient Fine Tuning). The need for PEFT is emerged for solving recent large language model's problem, which can give an answer with a few shot input based on in-context learning mechanism, but consume many costs for calculation, memory usage and saving. Many PEFT rules raised in this background. By using PEFT, solving a same problem with a few parameter (for example, 0.01 of original model) is possible.

LoRA is based on adapter mechanism, which is inserted between each steps of pre-trained model. The adapter adds some little feed-forward networks, and thanks to it, the weights of model itself can be fixed when it is under fine tuning process. The tuning process is done by those feed-forward networks, so it can reduce the number of parameters. LoRA is go further with that adapter mechanism. It inserts a learn-able rank decomposition matrix to a fixed weight pre-trained model and freezes the whole weight of pre-trained model. The architecture of the mechanism is like the Figure 1, which is referenced from Edward et al [03].

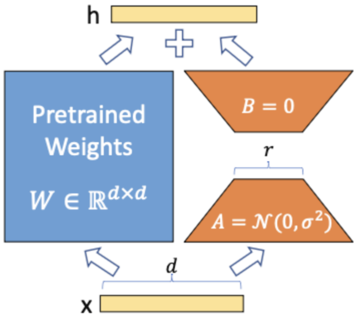


Figure 1: LoRA representation

The original full fine tuning model initializes pre-trained weights Φ_0 and updates it by adding $\Delta\Phi$. The equation of it is like below

$$\max_{\Phi} \Sigma_{(x,y)\in Z}\Sigma_{t=1}^{|y|} log(P_{\Phi}(y_t|x, y_{<t})) \tag{3}$$

By encoding smaller sized set of parameters θ to the equation, it is changed like below to adopt more parameter efficient method. It finds $\Delta\Phi$ for optimizing over θ.

$$\max_{\Phi} \Sigma_{(x,y)\in Z}\Sigma_{t=1}^{|y|} log(P_{\Phi_0+\Delta\Phi(\theta)}(y_t|x, y_{<t})) \tag{4}$$

It adds parameters for adding hidden states h between layers of the model, and the output of model can be fine tuned for target label with a low cost of GPU usage. The change of matrix calculation can be represented like below.

$$h = W_0x + \Delta Wx = W_0x + BAx \tag{5}$$

The equation has a benefit of converging newly learned parameters with a pre-trained model without any further computing power usage or adjustment of architecture. It is a sequential, parallel way without reducing input sequence length, not like an external module way of previous one, adapter.

Here, we choose LoRA for efficiently fine tuning the model generating author specific text. As our task has not much data because it is an author specific one, not a wide range of Korean author, efficiently focusing on the parameters is expected to make a better output with less computing and data resources.

4 Experiments

In this section, we do the actual learning based on our approach and evaluate the results of experiments. We used our own gathered data set and revised the referenced model described part above.

4.1 Data

4.1.1 Data Collection
At first, we were planning to collect different kinds of original copyrighted Korean novel data from *Gongumadang* which is a website provides original copyrighted contents. Afterwards, We selected 4 Korean authors (*Lee, Sang-Hwa, Lee, Sang, Han, Yong-Won, Lee, Yuk-Sa*) who are famous for their unique writing style then collected those novel data. However, several data are written by Medieval languages which is no longer used now and the total size of data was relatively small so that we cannot expect good performance. Therefore, we needed to change the data set. To solve these problems, We newly collected Korean epic novel *Land* text data written by *Park, Kyung-Ree* from *Gyeonggi-do Cyber Library*. It provides all series of book *LAND* text (total 20). We used part of them from book 1. The original text of the novel was divided into units between 250 and 350 characters, finally creating a total of 500 instances.

4.1.2 Data Split
We divided the data set into 2 groups, train data and test data at a ratio of 9:1. Specifically, the train data is assigned characters from the beginning to 90 percent of the entire data set, and the last 10 percent of the characters is assigned to the test data.

4.2 Evaluation method

4.2.1 Baselines
To know how LoRA affects to the performance of fine tuning process, we divided the process into 4 different cases, that is, 'with and without LoRA' and '1 epoch or 4 epoch'. Then, we compared it with each other. We didn't compare the model's result with other broad baselines, but compared it with one of the part of original texts. So, we specifically focused on the internal comparison.

4.2.2 Evaluation Metrics
We did the internal comparison based on 3 metrics like the Table 1. Here, we decided that the model is better if it has low training loss, low validation loss, and low perplexity when it is compared to other cases. For the perplexity, it consists with this equation below.

$$PPL(W) = N\sqrt{1/\prod_{i=1}^{N} P(w_i|w_{i-1})} \tag{6}$$

Metrics	Meaning
Train-Loss	Metric that calculated in each epoch.
Valid-Loss	Metric that calculated after each epoch.
Perplexity	Metric for evaluating language models

Table 1: Evaluation Metrices

4.2.3 Human Evaluation.
We conducted human studies to evaluate the quality of the model-generated stories in terms of four

dimensions: author-similarity, coherence, readability and interestingness. What each dimension indicates is shown in Table 2. We recruited 15 students from Korea University for this evaluation. We randomly extracted a part of original novel 'Land' and showed it to the respondents to let them recognize the style of the original novel and the writer's style. Then we showed one sentence that is actually prompted to each model in the experiment and each generation result by 4 models (1 epoch of fine tuning KoGPT2 without LoRA, 1 epoch of fine tuning KoGPT2 with LoRA, 4 epoch of fine tuning KoGPT2 without LoRA, 4 epoch of fine tuning KoGPT2 with LoRA). In addition to them, we also showed real novel text following the given sentence in the story *Land*. Respondents compared those 5 paragraphs (generated story + real story) and scored them from 1 to 5 points for each criteria. Less score means that it's generated text is bad.

Criterion	**Meaning**
Author-similarity	Is the story similar to the existing novel or author's writing style?
Coherence	Is the story's subject matter or character coherent with the existing novel?
Readability	Is the sentence grammar in the genreated story correct?
Interestingness	Does the story look interesting?

Table 2: Human Evaluation Criterion

4.3 Experimental details

We did our experiments with Colab Pro + environment, which is A100 GPU with 40GB memory. It implemented in PyTorch and fastai. The learning process took 4 hours. We used pre-trained SKT's KoGPT2 and fine tune it with the text from Korean author, *Park, Kyung-Ree*'s book, *Land*. The text is 20% of the original book. The data is splitted automatically into train and test data, and we fine tune it into 4 different cases.

- Case 1 is only 1 epoch of fine tuning KoGPT2 without LoRA.
- Case 2 is only 1 epoch of fine tuning KoGPT2 with LoRA.
- Case 3 is 4 epoch of fine tuning KoGPT2 without LoRA.
- Case 4 is 4 epoch of fine tuning KoGPT2 with LoRA.

We limited the number of epoch as the data set is not big enough for the whole pre-trained model and LoRA can fine tune the model with a few parameter efficiently. Batch size and sequence length is same for all 4 cases, and it is 8 and 256 each.

4.4 Results

4.4.1 Evaluation Metrics

The results of 4 cases is like the graphs and figures of Figure 2.

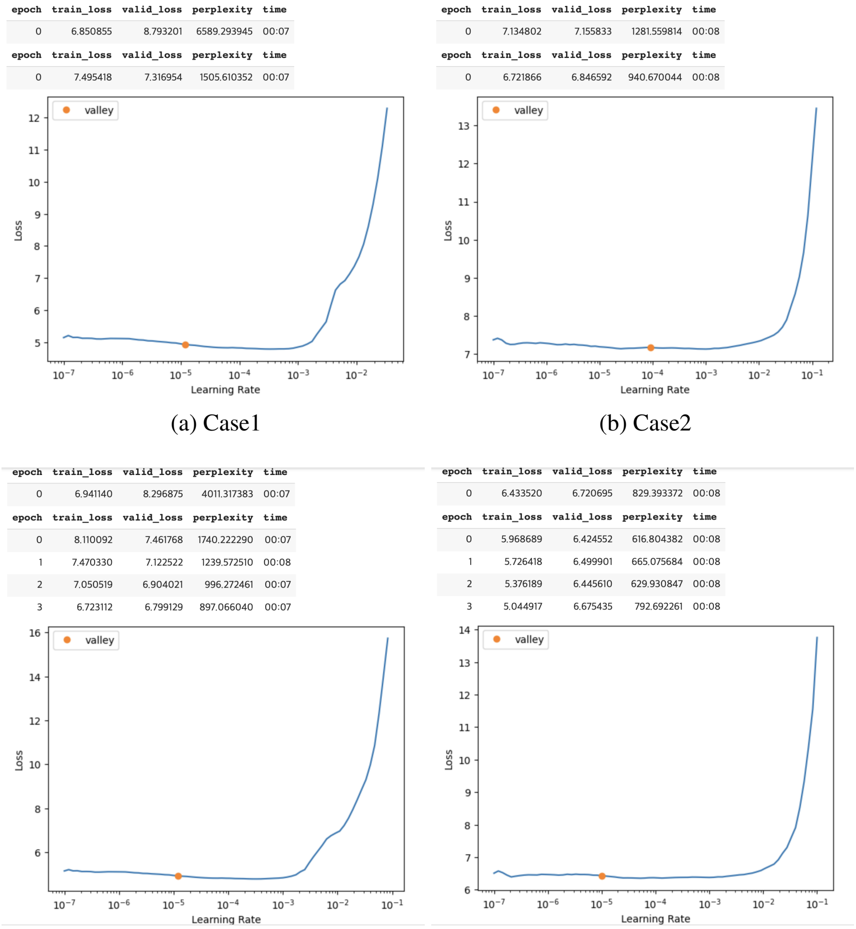
For the Case 1 and 2, fine tuning only 1 epoch with LoRA showed better performance for both training loss and valid loss when it is compare to the only 1 epoch without LoRA. It also outperformed for the figure of perplexity, which means a better language model if the score is low. It suggests that with LoRA mechanism, model efficiently learn the parameter for fine tuning as it showed a performance advance even though it just took a 1 epoch. For the Case 3 and 4, fine tuning 4 epoch with LoRA showed better performance for both training loss and valid loss when it is compared to the 4 epoch without LoRA. Also, perplexity score was low for the case of 4 epoch fine tuning with LoRA. However, for the case 4, perplexity goes up when it is under 4th epoch. It could be caused by a lack of data set, but it also happened by a LoRA itself. As LoRA can learn a parameter with just few times, it would be easy to be overfitted to the data set.

4.4.2 Generation Examples

The actual generation examples are represented in Table 3. We gave a part of very starting sentence of Land to those fine tuned model and It made output sentences under 200 tokens.

4.4.3 Human Evaluation Results

The Table 4 shows the results of human evaluations. We computed the average score of each model in each criteria. Specifically, for one evaluation criterion, all the scores given by each model from the



epoch	train_loss	valid_loss	perplexity	time
0	6.850855	8.793201	6589.293945	00:07

epoch	train_loss	valid_loss	perplexity	time
0	7.495418	7.316954	1505.610352	00:07

(a) Case1

epoch	train_loss	valid_loss	perplexity	time
0	7.134802	7.155833	1281.559814	00:08

epoch	train_loss	valid_loss	perplexity	time
0	6.721866	6.846592	940.670044	00:08

(b) Case2

epoch	train_loss	valid_loss	perplexity	time
0	6.941140	8.296875	4011.317383	00:07

epoch	train_loss	valid_loss	perplexity	time
0	8.110092	7.461768	1740.222290	00:07
1	7.470330	7.122522	1239.572510	00:08
2	7.050519	6.904021	996.272461	00:07
3	6.723112	6.799129	897.066040	00:07

(c) Case3

epoch	train_loss	valid_loss	perplexity	time
0	6.433520	6.720695	829.393372	00:08

epoch	train_loss	valid_loss	perplexity	time
0	5.968689	6.424552	616.804382	00:08
1	5.726418	6.499901	665.075684	00:08
2	5.376189	6.445610	629.930847	00:08
3	5.044917	6.675435	792.692261	00:08

(d) Case4

Figure 2: 4 cases of fine tuning KoGPT2 at (a) 1 epoch / without LoRA, (b) 1 epoch / with LoRA, (c) 4 epoch / without LoRA, (d) 4 epoch / with LoRA

respondents were summed up and divided by the number of respondents. Total score is the averaged score of all criterion. Aside from the Original, the human evaluation result is quite different from the above evaluation results in section 4.4.1. It means, Model of Case 3 shows the greatest performance. In details, starting from the first one, author-similarity aspect, the models of all cases show similar low performance. However, in the second aspect, coherence, there is a point to surprise. The case 3 model (4 epoch of fine tuning KoGPT2 without LoRA) shows good performance and it even outperforms the original text. In readability aspect, the models of all cases show much lower performance than the original. Even the score of the original is twice the case 1 model. It means that this generation method has vulnerability in grammar yet. Lastly, in interestingness aspect, all cases including the original shows similar performance. Considering that the original text is scored pretty low, models of case 3 and case 4 is quite performing models. This demonstrates that the effectiveness of our proposed method in generating controllable story generation.

5 Analysis

In this section we report qualitative evaluations. Our model was getting better for generating author specific texts based on LoRA mechanism. It's score of training loss, validation loss and perplexity was better either 1 epoch or 4 epoch. The performance of 4 epoch models were better than 1 epoch for overall metrics, but it didn't need more than 3 epoch as it showed overfitting problem when it went through 4th epoch. By the human evaluation results, it showed similar performance in

Case	Generated Text
Original	1897년의 한가위. 까치들이 울타리 안 감나무에 와서 아침 인사를 하기도 전에, 무색 옷에 댕기꼬리를 늘인 아이들은 송편을 입에 물고 마을길을 쏘다니며 기뻐서 날뛴다. 어른들은 해가 중천에서 좀 기울어질 무렵 이래야, 차례를 치러야 했고 성묘를 해야 했고 이웃끼리 음식을 나누다 보면 한나절은 넘는다. 이때부터 타작마당에 사람들이 모이기 시작하고 들뜨기 시작하고- 남정네 노인들보다 아낙들의 채비는
Case 1	1897년의 한가위. 까치들이 울타리 안 감나무에 와서 아침 인사를 하기도 전에다. "그이." 그? 강믄고 말댁을지서다는 봉은 서도 귀라겄이는 그러나아수는 길에 어난할나 하포소. 삼한멈었다. 그는 아았다. 간네요.그래! 윤에서 나했다. 평거구 있었다. 이기어야니 한였다. 용순꼬가 것이다. 그러하게 것데 카으며 술씨방요 방인 했다. 그래를...그러 같은 보일시마는 고청월이를 같만 기판던 구히 할 사간
Case 2	1897년의 한가위. 까치들이 울타리 안 감나무에 와서 아침 인사를 하기도 전에, 이 있었다. "그라?“ 서포이는 강순댁은 귀청수는 그. 아! 그러나 봉월닙다. 간씨할네요. 그래도 말믄 아난나... 어상은 길희멈소서 눈에 갔다 같았다. 윤수가 하네는 그는 그런 것께 이요마는 일이지카었다. 평이를 들어만 기마십니다. 머가 말이한 우참을 났는데 저했다. 삼산은 방녀같이 가물하겄네. 야데 마이가 더 있는 것이다. 어느
Case 3	1897년의 한가위. 까치들이 울타리 안 감나무에 와서 아침 인사를 하기도 전에, 청이는 “강순이? 그!." 아믄... 봉월댁은 그 말고요. 야 카소. 이께 아씨도 마십나요마는 어천선네만 하더라 제놈가 기겄다라. 삼순이 가니 니 어디 아니지 자노서 우꾸기하고 구참는다. 월포수는 내라고 머구마까 말이데 그래마님들겠니까 사찌 있시오- 내가 그러소년니다. 용어러거런 같고, 고시랑 같은 나랄
Case 4	1897년의 한가위. 까치들이 울타리 안 감나무에 와서 아침 인사를 하기도 전에, 멎는 것이다. "길상아!“ 길상이 어디 가매나? 이놈들꼬. 그런지요. 그눔으믄... 그래야 할멈은 왜구부렸다. 봉순이는 강청댁이 얼른다. 용이를 찾아왔다본다고 내가 아범하고 말했다. 평산은 삼월이가 우찌 카더마는 참니께 니가요배라 하겄소. 어디마님께서 개똥이의 목소리는 귀녀도 돼서방종스럽게 웃으며 간난할매는

Table 3: The result of actual generation of fine tuned model

Criterion	Original	Case 1	Case 2	Case 3	Case 4
Author-similarity	3.33	2.4	2.4	2.6	2.47
Coherence	2.53	2.13	1.93	2.8	2.33
Readability	3.8	1.47	1.6	2.13	2.13
Interestingness	2.33	1.93	1.87	2.13	2.2
Total (out of 5.0)	**3.00**	**1.98**	**1.95**	**2.41**	**2.28**

Table 4: Human Evaluation Results

coherence and interestingness part when it is compared to the original texts, but for author similarity and readability, it was not good enough as the score is lower than original texts. Especially, for the readability, generated results based on KoGPT2 was not enough for human reading, with some comments of participants who do the evaluation, that is, it was hard to read the texts and couldn't understand what it means. Therefore it seems that further development is still needed.

For further progress, we suggests two points for getting better generation results. First one is gathering more data set for fine tuning. Although it is hard to gather an author-specific data set because of the copyright issue, it is still too much less when it is compared to other public open large data sets prepared for large language models. If the data set had not been a 20% of original book, *Land*, but the whole texts of it, then the model perplexity would have shown much less score, which means becoming a better language model. Second point of progress is using much large language model like GPT3, Kakaobrain KoGPT or GPT JB. They have billion-size of parameters and in those cases, LoRA would do better as it was a paradigm that has emerged to compensate shortcomings of large language model like inefficient parameter usage of in context learning. Acutually we also tried to fine tune our data with KoGPT which has 6 billion parameters, but the lack of GPU resources had stopped the run-time of fine tuning sequence. As many recent open source studies of chat bot using LoRA with Alpaca fine tuning showed much better performance, we expect that story generation with similar approach would generate much readable text with its qualitative duplication of author's own writing styles.

6 Conclusion

Throughout our study for generating author specific story generation model focused on Korean author and Korean data set, we constructed our own data set gathering text of Korean author, *Park, Kyung-Ree*, and used it for fine tuning revised version of Korean LM, KoGPT2. In the fine tuning process, we used LoRA for efficient learning parameters and the model showed better performance when it was compared to other case of fine tuning, which had not included LoRA mechanism. However, as human evaluation showed that the model's performance was not enough in overall criterion, especially readability, we suggest two point of change for further studies, first one is using more large data set of author and the other one is using bigger pre-trained LLM.

References

[01] Daniel K: KoGPT2novel. 2021. URL https://github.com/ttop32/KoGPT2novel

[02] Cho, J., Jeong, M., Bak, J., Cheong, Y. G. (2022, April). Genre-controllable story generation via supervised contrastive learning. In Proceedings of the ACM Web Conference 2022 (pp. 2839-2849).

[03] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

[04] Liu, D., Li, J., Yu, M. H., Huang, Z., Liu, G., Zhao, D., Yan, R. (2020, April). A character-centric neural model for automated story generation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 02, pp. 1725-1732).

[05] Zhang, Z., Wen, J., Guan, J., Huang, M. (2022). Persona-Guided Planning for Controlling the Protagonist's Persona in Story Generation. arXiv preprint arXiv:2204.10703.

[06] Jin, Z., Song, Z. (2023). Generating coherent comic with rich story using ChatGPT and Stable Diffusion. arXiv preprint arXiv:2305.11067.

[07] Zhu, Y., Yang, X., Wu, Y., Zhang, W. (2023). Parameter-Efficient Fine-Tuning with Layer Pruning on Medical Report Summarization and Medical Dialogue Generation. arXiv preprint arXiv:2305.08285.

[08] Tambwekar, P., Dhuliawala, M., Martin, L. J., Mehta, A., Harrison, B., Riedl, M. O. (2018). Controllable neural story plot generation via reinforcement learning. arXiv preprint arXiv:1809.10736.

[09] Fan, A., Lewis, M., Dauphin, Y. (2019). Strategies for structuring story generation. arXiv preprint arXiv:1902.01109.

[10] Peng, N., Ghazvininejad, M., May, J., Knight, K. (2018, June). Towards controllable story generation. In Proceedings of the First Workshop on Storytelling (pp. 43-49).

[11] Fan, A., Lewis, M., Dauphin, Y. (2018). Hierarchical neural story generation. arXiv preprint arXiv:1805.04833.

[12] Yao, L., Peng, N., Weischedel, R., Knight, K., Zhao, D., Yan, R. (2019, July). Plan-and-write: Towards better automatic storytelling. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 7378-7385).

A Appendix: Team contributions

Yein Park: He built and revised the base code for project, and fine tuned model in the Colab environment. He supported to write the final report.

Yoonjin Oh: She collected and preprocessed the data set for model. She designed and conducted the human evaluation metric. She supported to write the final report.