Finding and Generating a Missing Part for Story Completion

Yusuke Mori[1] **Hiroaki Yamane**[2,1] **Yusuke Mukuta**[1,2] **Tatsuya Harada**[1,2]
[1]The University of Tokyo, [2]RIKEN
{mori, mukuta, harada}@mi.t.u-tokyo.ac.jp
hiroaki.yamane@riken.jp

Abstract

Creating a story is difficult. Professional writers often experience a writer's block. Thus, providing automatic support to writers is crucial but also challenging. Recently, in the field of generating and understanding stories, story completion (SC) has been proposed as a method for generating missing parts of an incomplete story. Despite this method's usefulness in providing creative support, its applicability is currently limited because it requires the user to have prior knowledge of the missing part of a story. Writers do not always know which part of their writing is flawed. To overcome this problem, we propose a novel approach called "missing position prediction (MPP)." Given an incomplete story, we aim to predict the position of the missing part. We also propose a novel method for MPP and SC. We first conduct an experiment focusing on MPP, and our analysis shows that highly accurate predictions can be obtained when the missing part of a story is the beginning or the end. This suggests that if a story has a specific beginning or end, they play significant roles. We conduct an experiment on SC using MPP, and our proposed method demonstrates promising results.

1 Introduction

Currently, because of the Internet, anybody can freely publish their original stories. However, it is challenging to write something that people would like and want to read. Sometimes, even professional writers fall into slumps during the writing process.

Numerous studies on understanding the secret of creating good stories have been conducted (Campbell, 1949; Propp, 1968). Rules for creating stories have been studied extensively, and "three-act structure" (Field, 2006) and "Save the cat" (Snyder, 2005) are popular examples. These works can help guide people who want to write good stories to demonstrate their creativity.

With the development of machine learning (ML) and natural language processing (NLP) technology in recent years, the creation of an automated system that supports the creative endeavors of people is now feasible (Roemmele, 2016; Peng et al., 2018; Yao et al., 2019; Goldfarb-Tarrant et al., 2019). To assist people in creating stories, it is essential to train computers to understand and create stories.

To measure the reading comprehension abilities of systems regarding stories, Mostafazadeh et al. (2016) proposed the "Story Cloze Test" (SCT). In the SCT, four sentences are presented, and the last sentence is excluded from an original five-sentence story. The objective of this task is to select an appropriate sentence from two options that complement the missing last sentence. Based on this approach, Wang and Wan (2019) proposed the "Story Completion (SC)" task in the field of generating and understanding stories. Given any four sentences of a five-sentence story, the objective of this task is to generate the sentence that is not given (known as the missing plot) to complete the story.

The ability to solve the SC is essential in the context of creative support. If writers cannot complete a story or a plot, a suitable model can provide them with the appropriate support. However, such applications are currently restricted because they require the user to know which part of a story is missing in advance. When considering an actual application, writers do not always know where their writing is

Proceedings of LaTeCH-CLfL 2020, pages 156–166
Barcelona, Spain (Online), December 12, 2020.

flawed, as evidenced by the vital role of editors who work with them. Of course, the editors do not just point out the missing points; they play various roles. For example, they also point out unnecessary parts. Keeping this in mind, let us understand one of the roles of the editor.

To overcome this limitation, we propose a new story comprehension method named "Missing Position Prediction (MPP)," as shown in Figure 1. An incomplete story with one sentence missing is given as input. Unlike in the SC, no information regarding the position of the missing content is given. This task aims to predict the position of the missing part. The ability to solve this task indicates that computers can identify flaws in a story's plot. In practical applications, by combining our novel task with an appropriate SC method, writers can benefit from computerized completion even if they do not know where a flaw is.

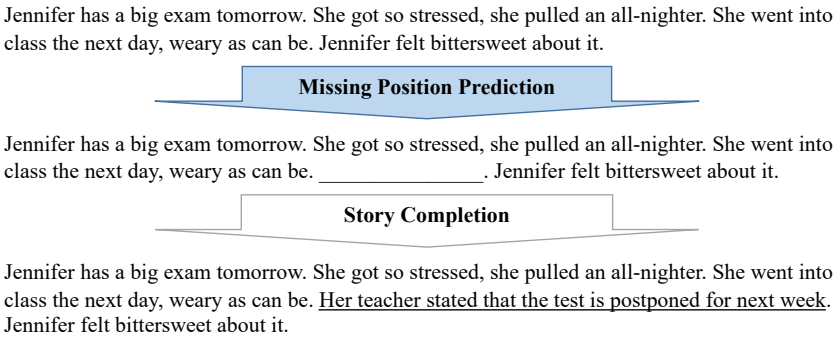


Figure 1: Example of an incomplete story and flow of MPP and SC.

Additionally, we propose a novel method for MPP and SC. Given an incomplete story, it estimates the missing part and generates a sentence to complete the story. We make our code available to support further progress on our proposed task and SC.[1]

Our main contributions are as follows:

- We propose "Missing Position Prediction (MPP)" as a story comprehension method. This method predicts the position of a missing part of an incomplete story and has significance in the contexts of story understanding, story generation, and story-writing assistance.
- We propose a novel method for MPP and SC. We first perform an experiment focusing on the MPP, and our proposed method demonstrates promising results. An analysis of the results shows that highly accurate predictions can be obtained when the missing part of a story is its beginning or end.
- Based on the results of the MPP experiment, we conduct another experiment on SC using MPP. The results of the experiment show that given an incomplete story, it is possible to restore it such that it is comparable with the original human-written one.

2 Related Work

2.1 Reading Comprehension on Stories

In some studies, to better comprehend stories, the stories were considered to be collections of events. The "Narrative Cloze Test" (Chambers and Jurafsky, 2008) is a typical example. Mostafazadeh et al. (2016) proposed the SCT as a more difficult task. The SCT presents four sentences, and the last sentence is excluded from a story composed of five sentences. The system must select an appropriate sentence from two choices that complement the missing last sentence. In addition to the task, the authors released a large-scale story corpus named "ROCStories," which is a collection of non-fictional daily-life stories written by hundreds of workers belonging to Amazon Mechanical Turk (Amazon MTurk).

In our proposed task, it is essential to understand the remaining information to infer what is missing. Regarding the example in Figure 1, the third sentence states that Jennifer is weary, and the fourth sentence mentions that she felt bittersweet. It is estimated that something mentioned as "it" is missing, and "it" is

[1] https://github.com/mil-tokyo/missing-position-prediction

the reason for her change of feeling. In this manner, it is necessary to identify unnaturalness – that is, the parts where the narrative arc is broken – in a story. This is a more challenging task than SCT. We believe this task is deeply related to a fundamental question in story understanding: whether or not the model understands the flow of a story.

2.2 Partial Generation of Stories

Inspired by the SCT, Zhao et al. (2018) designed "Story Ending Generation (SEG)" as a subtask of story generation. Given an incomplete story, where the last sentence is excluded from the original five-sentence story, the objective of this task is to generate the last sentence, not to select. Furthermore, based on SEG, Wang and Wan (2019) proposed the SC and investigated the problem of generating a missing story plot at any position in an incomplete story.

Additionally, in recent years, research regarding text infilling has been actively conducted (Ippolito et al., 2019; Donahue et al., 2020; Huang et al., 2020). Regarding stories, Ippolito et al. (2019) worked on complementing the missing span between left and right contexts, which they called "story infilling." In the appendix, they reported that they tried human evaluation using Amazon MTurk but their task was too hard for the average worker. Although there is no mention of why the task was too hard for the average worker, we suspect that the length of the text in their task may have been one of the reasons.

These studies require a writer to have prior knowledge of the missing parts and do not consider the case where the writer is unaware of the flaws in his/her work. The MPP aims to fill this gap.

We should note that even when there is a missing part in the story, it may be caused by writer's intention that "I want the readers to read between lines". However, the missing part can also be an unintentional mistake. To analyze if the model can understand whether the missing part is an "writer's intentional missing" is out of the scope of this study. At this stage, MPP is especially effective in the latter case, unintentional mistake. However, when a model's understanding of writer's intentional missing is achieved, it is expected that writers can also be benefited in the former case – using a method of MPP, they can know whether their intention is well understandable by readers.

As the first step, we used short stories for this task. Instead of asking average workers, we did a qualification test and only qualified workers could participate in the evaluation.

2.3 Seq2seq for Text Generation

In SEG, a simple type of the sequence-to-sequence model (Seq2seq) (Sutskever et al., 2014) and an extension using the attention mechanism are typically used as baselines (Zhao et al., 2018; Li et al., 2018; Guan et al., 2019; Mori et al., 2019).

The use of unsupervised pre-trained large neural models, such as BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), has become mainstream in NLP. BERT is originally trained as a masked language model and considered unsuitable for text generation compared with models using a left-to-right architecture, such as GPT-2, XLNet (Yang et al., 2019), and BART (Lewis et al., 2020). However, experiments conducted by Rothe et al. (2020) using BERT and GPT-2 for Seq2seq demonstrated interesting results. Although they did not claim that BERT is optimal as a decoder, they demonstrated that BERT2BERT outperforms BERT2GPT in some generation tasks.

In this study, we extend the Seq2seq-based method for SEG to solve MPP and SC. To achieve a more natural sentence completion, we use BERT as a Seq2seq decoder and BERT-derived Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) as a part of an encoder.

3 Task Description

We begin by formulating SEG and the SC, after which we formulate our proposed task.

3.1 Story Ending Generation and Story Completion

We define $S = \{s_1, s_2, ..., s_n\}$ as a story comprising n sentences. In SEG, $S' = \{s_1, s_2, ..., s_{n-1}\}$ is given as an input. The objective of the task is to generate an appropriate ending. For SC, an incomplete story consisting of $n-1$ sentences $S' = \{s_1, ..., s_{k-1}, s_{k+1}, ..., s_n\}$, where k represents the position of

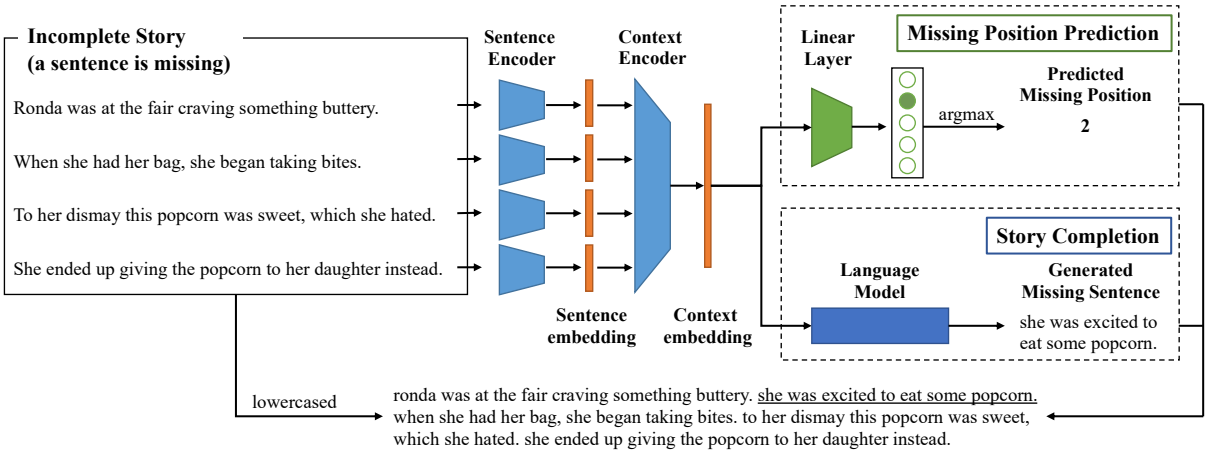


Figure 2: An overview of the proposed method.

the missing sentence in the story, is given. Next, we focus on the objective of the task that involves generating an appropriate sentence which is coherent with the given sentences. During each task, the model is trained to maximize probability $p(y|S')$, where y represents the ground truth sentence. Specifically, $y = s_n$ in SEG and $y = s_k$ in SC.

3.2 Missing Position Prediction

To overcome the issue whereby the SC model requires information regarding k, i.e., the position of the missing sentence, we propose the MPP to predict k from a given $n-1$ sentences, as shown in Figure 1. Similar to the SC, an incomplete story comprising $n-1$ sentences $S' = \{s_1, ..., s_{k-1}, s_{k+1}, ..., s_n\}$ is given as an input. However, any information regarding k is not given. The order of the sentences is known, but the missing position is unknown. Specifically, s_{k-1} and s_{k+1} are treated as continuous sentences. Our objective is to predict k from the input. In other words, the model is trained to maximize probability $p(missing = k|S')$.

4 Proposed Method

Hierarchical approaches have demonstrated effectiveness in story generation (Fan et al., 2018; Ravi et al., 2018). We propose a novel method with a hierarchical architecture for the MPP and SC. We devise a method inspired by the two-step encoder of Hierarchical-Seq2seq, which is a simple method for SEG that we proposed in our previous study (Mori et al., 2019). The first encoder receives $S' = \{s_1, ..., s_{k-1}, s_{k+1}, ..., s_n\}$ and outputs the sentence embeddings $\{v_1, ..., v_{k-1}, v_{k+1}, ..., v_n\}$. Next, the second encoder receives the sentence embeddings and generates a distributed representation of the entire context $v_{context}$. We call the first encoder "sentence encoder," and the second encoder "context encoder." For MPP, we input $v_{context}$ into a linear layer and obtain a five-unit output. For SC, we input $v_{context}$ into a language model and obtain a sentence to complete the story. Figure 2 shows an overview of the proposed method. Although the output of the MPP can be used here, we prefer to have our model learn these two tasks simultaneously. We intend to take it up as future work to use predicted MPP for SC and vice versa.

In proposing a new task, we believe it is useful to test how well a simple method can solve the task. Analyzing the performance and characteristics of a simple method will help in the application of complex methods in future studies. Thus, we propose a new task along with a simple method.

4.1 Sentence Encoder

First, we obtain sentence embeddings v_j for each input sentence s_j in a given context. We apply SBERT in each sentence. This encoder is not fine-tuned during our training.

4.2 Context Encoder

Using the sentence embeddings obtained, we apply another encoding layer to handle context embedding $v_{context}$. Although there is a discontinuity in the input and the missing position k is not given, the order is preserved. Hence, it is considered to be appropriate to treat the input as a sequence. We propose to use gated recurrent unit (GRU) (Cho et al., 2014) as the main part of the encoder. GRU is a type of RNN and is useful in handling sequences. Ravi et al. (2018) used an RNN with GRU cells for story encoding, demonstrating that a GRU is sufficient to capture the sequence in a short story. Li et al. (2019) also used GRU and its variant in their Context Encoder. The output of the GRU is input into a linear layer and a batch normalization layer (Ioffe and Szegedy, 2015).

4.3 Language Model

We use BERT as a language model for generating sentences to fill in the missing parts. Here, BERT is used as a decoder, and the output of the context encoder is used to initialize the encoder hidden states in cross-attention. Starting from the start token, we repeat the next token prediction for sentence generation.

5 Experiment 1: Missing Position Prediction

First, we worked on learning the MPP only. In this experiment, we investigated the part of the proposed method that excludes the language model.

5.1 Dataset

set	#stories	missing position
train	78,528	Given randomly during training
dev	9,816	Given when creating dataset
test	9,817	Given when creating dataset
total	98,161	

Table 1: Overview of the dataset used.

ROCStories is a well-organized corpus and is widely used in story-generation tasks; it is typically used in SEG (Zhao et al., 2018; Li et al., 2018; Guan et al., 2019). Similarly, Wang and Wan (2019) used it for their story-completion task. Furthermore, the dataset was used by Peng et al. (2018) for controllable story generation. Qin et al. (2019) tackled "Counterfactual Story Rewriting," which is a story revising task, using their proposed TIMETRAVEL dataset built using ROCStories. Although, initially, we did consider using other datasets, such as WritingPrompts, we ultimately did not use them. Stories in WritingPrompts vary in terms of length, and therefore, the importance of a single sentence varies from one story to the other. Thus, considering the requirements of our analysis, the aforementioned dataset seemed inappropriate.

Thus, as a starting point for proposing the task, we used ROCStories. As shown in Table 1, the dataset was randomly split in the ratio of 8:1:1 to obtain the training, development, and test sets, respectively. We removed one sentence from a five-sentence story. The missing position k was randomly decided based on a discrete uniform distribution. For the development and test sets, this removal procedure was performed when creating the dataset to improve reproducibility. For the training set, we retained the original five-sentence story in the dataset and removed a sentence randomly when reading the data during training. As a result, a different sentence could be removed from the same story, with a different k value, thus acting as data augmentation.

5.2 Comparison Method

Max-pool Context. To examine the usefulness of treating context as a sequence in the proposed task, we trained another model. In this setting, a max pooling layer was used as a context encoder.

5.3 Training Details

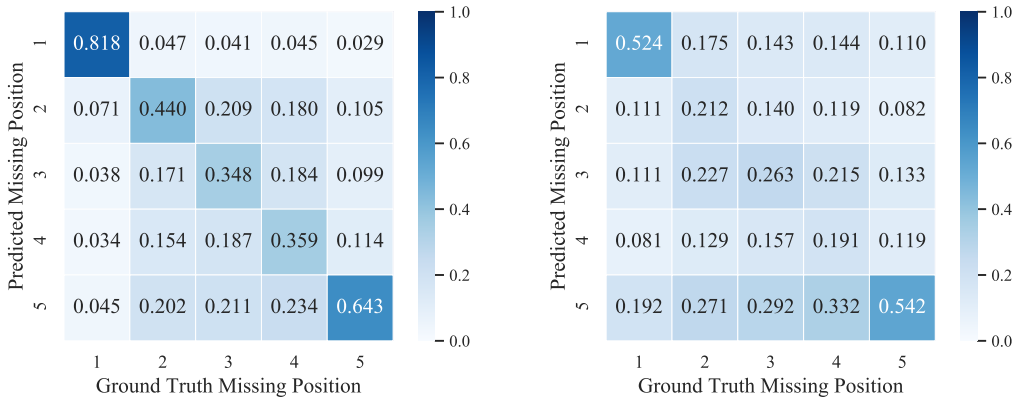
We trained a model for 30 epochs. The validation loss for every epoch was calculated, and the state with the smallest validation loss was used for further tests. Among the trained SBERTs, we used "bert-base-nli-mean-tokens." The output dimension was 768. For the GRU context, the number of hidden units of the GRU was 256. The linear layer had 256 dimensions for both the input and output, and weights were initialized from a normal distribution with $mean = 0$, $std = 0.01$. For the max-pool context, we applied max pooling to sentence embeddings and obtained a vector with the same dimension as the sentence embedding. We then input this vector into a linear layer and obtained a 256-dimensional vector as the context vector. The linear layer for receiving the output of the context encoder and for identifying the five labels had a 256-dimensional input and a five-dimensional output. We used the Adam optimizer with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0. A gradient clipping with a value of 5 was used. We set the batch size to 256.

5.4 Results

For each method, we performed five trials while changing the random seed at the time of training and calculated the mean and standard deviations of the accuracy. As shown in Table 2, the GRU context achieved an accuracy of $52.2 \pm 0.220\%$, which was higher than the accuracy of the max-pool context. The results indicated the usefulness of treating context as a sequence in the proposed task.

Methods	Accuracy (%)
Max-pool Context	35.0±0.334
GRU Context	**52.2±0.220**

Table 2: Prediction accuracy, shown as $mean \pm std$. It is a five-class classification task, so the chance rate is 20%.



(a) GRU context (b) Max-pool context

Figure 3: Heat maps showing the results of the (a) GRU context and (b) Max-pool context. The ground truth (GT) label is shown on the x-axis and the predicted label is on the y-axis. The squares on the diagonal line denote correct cases. The ratios of the predicted label to the GT label are shown numerically.

Hereinafter, for a more detailed discussion, we use one of the five trials as an example. The heat map in Figure 3 (a) shows which positions can be accurately identified using the GRU context method. When the sentence 1 was missing, the accuracy exceeded 80%. The results of sentences 2 to 4 exhibited lower accuracy, whereas sentence 5 had a higher accuracy.

Figure 3 (b) shows the max-pool context result for each missing position. Even though this method does not consider the sequence of the context, the prediction results for the sentences 2 to 4 are lower than those for sentences 1 and 5. Thus, it can be inferred that treating a context as a sequence does not adversely affect the prediction of missing middle sentences.

6 Experiment 2: Missing Position Prediction + Story Completion

Based on the results of Experiment 1, we conducted another experiment in which we tackled both MPP and SC. As a context encoder, we used the GRU context. We used the same dataset as in Section 5.1.

6.1 Training Details

We trained a model for 50 epochs. The validation loss for every epoch was calculated, and the state with the smallest validation loss was used for human evaluation. For the GRU context, the number of hidden units was 768. The linear layer had 768 dimensions for both the input and output. We used HuggingFace's implementation of BERT and its pre-trained model "bert-base-uncased" (Wolf et al., 2019). We calculated the total loss as follows: $L_{total} = 0.5 * L_{MPP} + 0.5 * L_{SC}$, where L_{MPP} represents the softmax cross entropy loss for MPP, and L_{SC} represents the softmax cross entropy loss for SC. We optimized the value of L_{total} using the AdamW optimizer with a learning rate of $1e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$, and a weight decay of 0. We used a linear learning rate warmup with 4 epochs. We used a gradient clipping with a value of 1 and set the batch size to 128.

6.2 Human Evaluation

We conducted human evaluation with the help of Amazon MTurk workers. We conducted two types of tasks: a qualification test and a pair-wise evaluation task.

To choose workers with a high degree of ability to evaluate stories for participation in the evaluation task, we first conducted a qualification test. Ten randomly selected questions from the validation set of the SCT were solved by the workers, and only those workers who answered all ten questions correctly were allowed to participate in the next evaluation task.

For the pair-wise evaluation task, the qualified workers were given two similar short stories, and they were asked to choose which story gave the impression of being a complete story. The workers were given four choices as follows: Option A is more appropriate, Option B is more appropriate, both options are equally appropriate, and neither option is suitable. In this evaluation task, workers were also required to write the reason for their answer. We used 200 story pairs for comparison. The original human-written story (GT) is from the test set shown in Table 1, and our proposed model generated the other candidates based on an incomplete story. Five workers evaluated each story pair. Among the five answers obtained for each story, the most frequently chosen answers were considered as an agreement among the workers. Notably, that the workers did not do the same number of tasks. Therefore, instead of calculating the inter-annotator agreement, we decided to consider the most frequent answer.

6.3 Results

The results of the human evaluation are shown in Table 3.

Proposed	GT	both	neither
8	148	44	0

Table 3: Human evaluation results of pair-wise experiment. We used 200 stories, and each story was evaluated by five workers. The most frequently chosen answers were considered as their agreement.

Regarding the 200 stories that were autocompleted, eight were judged to be better than the original story, and 44 were judged to be equivalent to the original story. In other words, our proposed method can generate a story that is either as good as or better than a GT story with 26% probability.

7 Discussion

The results of Experiment 1 support the following two findings from Wang and Wan (2019): 1) The plot becomes more complicated as it progresses, thereby making the estimation of latter sentences more difficult and 2) for $k = 5$, four sentences in the context are continuous. Therefore, a good expression can be easily obtained, even by using an encoder that does not consider discontinuity. It is interesting to

note that the beginning or the end of a story can be predicted with the highest accuracy. This appears to be related to the fact that the collection of ROCStories was performed with the following in mind: "the story should read like a coherent story, with a specific beginning and ending." In other words, the story under consideration has a specific beginning and ending. Thus, if the beginning or the ending is missing, it can be interpreted as that the methods treat the story as particularly unnatural and predict the missing position with high accuracy.

For qualitative analysis on Experiment 2, we show three examples of story pairs and human evaluations in Table 4. In the first example, the autocompleted story was evaluated to be better than the GT. MPP was a success, and a contextualized completion sentence was generated. In the second example, the autocomplete story was rated as equivalent to the GT. MPP estimated a missing location that differed from the original story but increased information differently from the GT, which was appreciated by the workers. In the third example, autocompletion did not work. It succeeded in MPP, but it failed in generating a contextualized completion sentence. The failure to generate an essential word ("contest") is pointed out. Note that the second answer appears to have been mischaracterized.

8 Conclusion

To overcome the issue of conventional SC tasks that require information regarding the position of the missing part in a story, we proposed a MPP to predict the position based on the given incomplete story. Our proposed method demonstrated that treating the context as a sequence is useful for solving this new task. We examined the prediction accuracy for each missing position and found that a prediction is easier if the beginning or the end of a story is missing. Furthermore, we tackled the combined task of MPP and SC. We conducted a pair-wise human evaluation against a human-written story, for which our proposed method demonstrated promising results.

Because we limit the study to five-sentence stories, it is unlikely that humans make mistakes in the plot. However, humans may overlook plot imperfections when considering longer, more complex stories. Thus, checking such mistakes is part of the editors' job. We proposed the task in the context of creative support, but it also can be positioned in the context of narrative understanding. Planning a story requires a form of reasoning that can move backward as well as forward. That is why SC tasks have significant meaning in story understanding and generation, and our proposed task would be a better test of a model's abilities to understand the flow of a story.

For the sake of simplicity, we proposed a simple machine-learning-based method. However, using simple bag-of-words methods or part-of-speech analysis may be effective for our proposed task. Therefore, exploring the efficacy of using methods other than those based on machine learning is left as future work.

However, our proposed task poses specific limitations. In our task, it is known that there is a missing position in the input story, and that there is only one such instance. In reality, an input story may be complete, that is, k is null. Furthermore, there may be a case in which there are multiple missing positions, that is, a case in which k has multiple values. Although dealing with these constraints is left for future studies, it is conceivable to introduce a certainty factor for the missing prediction. For example, predicting that k is null when the certainty factor is low. Although we considered a constrained case of study, we believe that our proposed task is an important step toward assisting writers in the creation of stories.

Acknowledgements

We would like to thank Yusuke Kurose, Naoyuki Gunji, and Ryohei Shimizu for helpful discussions. This work was supported by JST AIP Acceleration Research Grant Number JPMJCR20U3 and JSPS KAKENHI Grant Number JP20H05556, Japan.

Context	since the questions were complicated, i was extremely nervous. despite believing that i've failed, i turned the exam in. the teacher handed the exams back to us the next day. i ended up receiving a b.	
GT	i took my class final in math today. since the questions were complicated, i was extremely nervous. despite believing that i've failed, i turned the exam in. the teacher handed the exams back to us the next day. i ended up receiving a b.	
Ours	my teacher gave us a test. since the questions were complicated, i was extremely nervous. despite believing that i've failed, i turned the exam in. the teacher handed the exams back to us the next day. i ended up receiving a b.	
	Answers with Reasons (A: GT, B: Ours)	
	both	Whether it is a class final or a given test, both stories are the same and therefore both complete.
	neither	both doesn't make sense
	Ours	A is jumbled and does not make sense. B is logically arranged as a story.
	Ours	In "A," it wouldn't make sense that a final exam was handed back in class the next day.
	Ours	B was more appropriate since it is having a continuous flow than A
Context	tom was at a local park. there was an egg hunt for the kids. tom decided to pick some eggs up. he enjoyed the treats in them.	
GT	tom was at a local park. it was easter. there was an egg hunt for the kids. tom decided to pick some eggs up. he enjoyed the treats in them.	
Ours	tom was at a local park. there was an egg hunt for the kids. tom decided to pick some eggs up. tom was able to get many eggs. he enjoyed the treats in them.	
	Answers with Reasons (A: GT, B: Ours)	
	both	both are complete sentences
	Ours	Option B is complete as it says that tom was able to get some eggs in the hunt.
	both	Both of them can be considered complete. Story A tells us it is Easter (and story B doesn't) while Story B tells us Tom picked many eggs (and story A doesn't). Both of those details could be removed and the stories would still be the same.
	Ours	The fact that he was able to gather some eggs was more complete than just deciding to pick up some eggs. Story A Easter gave a better time context but did not really add as much to the story since traditionally an egg hunt is held on Easter so the omission of that in Story B was made up for Tom being able to gather some eggs.
	both	Both stories have a starting, content and ending.
Context	timothy loved to dance. timothy didn't have much confidence in himself. it took everything he had to dance with all of his self doubt. everyone loved his dancing and he won the contest.	
GT	timothy loved to dance. there was a dance contest that was coming up soon. timothy didn't have much confidence in himself. it took everything he had to dance with all of his self doubt. everyone loved his dancing and he won the contest.	
Ours	timothy loved to dance. he decided to take dance lessons. timothy didn't have much confidence in himself. it took everything he had to dance with all of his self doubt. everyone loved his dancing and he won the contest.	
	Answers with Reasons (A: Ours, B: GT)	
	GT	Only B makes sense and a complete story.
	GT	A is more correct and arranged
	GT	Story A doesn't mention the contest which Timothy ends up winning, therefore misses an important piece of the story.
	GT	Story B mentions that there was a dance contest at the start and that he won it at the end. Story A only mentions a contest abruptly at the end making it seem out of place.
	GT	B is more good

Table 4: Examples of original and autocompleted stories, followed by answers and reasoning by MTurk workers. The GT was not originally lowercased, but it was lowercased in our pair-wise evaluation task to compare with autocomplete stories. Additionally, the context given to the model is not lowercased, but it is lowercased here to make it easier to compare with the GT and our proposed method.

References

Joseph Campbell. 1949. *The Hero with a Thousand Faces*. Pantheon Books.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 789–797, Columbus, Ohio, June. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734, Doha, Qatar, October. 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, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501, Online, July. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia, July. Association for Computational Linguistics.

Syd Field. 2006. *The Screenwriter's Workbook, Revised Edition*. Delta Trade Paperbacks.

Seraphina Goldfarb-Tarrant, Haining Feng, and Nanyun Peng. 2019. Plan, Write, and Revise: an interactive system for open-domain story generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 89–97, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Jian Guan, Yansen Wang, and Minlie Huang. 2019. Story ending generation with incremental encoding and commonsense knowledge. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6473–6480, Honolulu, Hawaii, January–February. AAAI Press.

Yichen Huang, Yizhe Zhang, Oussama Elachqar, and Yu Cheng. 2020. INSET: Sentence infilling with INter-SEntential transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2502–2515, Online, July. Association for Computational Linguistics.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, July. PMLR.

Daphne Ippolito, David Grangier, Chris Callison-Burch, and Douglas Eck. 2019. Unsupervised hierarchical story infilling. In *Proceedings of the First Workshop on Narrative Understanding*, pages 37–43, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July. Association for Computational Linguistics.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Generating reasonable and diversified story ending using sequence to sequence model with adversarial training. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1033–1043, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. 2019. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June.

Yusuke Mori, Hiroaki Yamane, Yusuke Mukuta, and Tatsuya Harada. 2019. Toward a better story end: Collecting human evaluation with reasons. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 383–390, Tokyo, Japan, October–November. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California, June. Association for Computational Linguistics.

Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. 2018. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49, New Orleans, Louisiana, June. Association for Computational Linguistics.

Vladimir IAkovlevich Propp. 1968. *Morphology of the Folktale (Translated by L. Scott)*. University of Texas Press.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5043–5053, Hong Kong, China, November. Association for Computational Linguistics.

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

Hareesh Ravi, Lezi Wang, Carlos Muniz, Leonid Sigal, Dimitris Metaxas, and Mubbasir Kapadia. 2018. Show me a story: Towards coherent neural story illustration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3980–3990, Hong Kong, China, November. Association for Computational Linguistics.

Melissa Roemmele. 2016. Writing Stories with Help from Recurrent Neural Networks. In *AAAI Conference on Artificial Intelligence; Thirtieth AAAI Conference on Artificial Intelligence*, pages 4311 – 4312, Phoenix, AZ, February. AAAI Press.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Blake Snyder. 2005. *SAVE THE CAT! The Last Book on Screenwriting You'll Ever Need*. Michael Wiese Productions.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Tianming Wang and Xiaojun Wan. 2019. T-CVAE: Transformer-based conditioned variational autoencoder for story completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 5233–5239. International Joint Conferences on Artificial Intelligence Organization, July.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5753–5763. Curran Associates, Inc.

Lili Yao, Nanyun Peng, Weischedel Ralph, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-Write: Towards better automatic storytelling. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 7378–7385, Honolulu, Hawaii, January–February. AAAI Press.

Yan Zhao, Lu Liu, Chunhua Liu, Ruoyao Yang, and Dong Yu. 2018. From plots to endings: A reinforced pointer generator for story ending generation. In *Proceedings of Natural Language Processing and Chinese Computing*, volume abs/1901.03459.