The Nectar of Missing Position Prediction for Story Completion

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

Abstract

The story creation by providing automatic support to writers is a challenging and crucial task. Recently, in the field of story generation and understanding, story completion has been proposed as an approach of generating missing parts of an incomplete story. Despite the usefulness of this approach in providing creative support, its applicability is limited. This limitation is owing to the prior knowledge requirement to the user regarding the missing part of the story. To overcome this limitation, we proposed a novel approach called "Missing Position Prediction" [MYMH20]. It is necessary to predict the position of the missing part in an incomplete story. Through our study, we found that the estimation accuracy when the first or last sentence of a short story was missing was higher than when the middle sentences were missing. In this study, we further analyze this result using sentence similarity and part-of-speech tagging.

1 Introduction

With the recent advancement of natural language processing technology, an automated system that supports the creative endeavors of humans is now feasible [Roe16, PGMK18, YPR+19, GTFP19]. From this viewpoint, we focused on the "Story Completion" (SC) task proposed by Wang and Wan [WW19] in the field of story generation and understanding. In SC, any four sentences of a five-sentence story are provided and the objective is to generate a sentence that fills the missing part. The ability to solve SC is essential for creative support for story writers. If a writer cannot complete a story, a suitable SC model can provide them with the appropriate backing. However, in SC's conventional task setting, prior knowledge of the missing part's position is required, and this is a limitation. The writers do not always know where their writing text flaws. We believe that pointing out flaws that writers themselves are not aware of is also an essential part of a writing support system. To overcome the limitation of SC, we proposed a story comprehension task called "Missing Position Prediction (MPP)" [MYMH20], as shown in Fig. 1. Wherein, an incomplete story is input without information about the position of the missing part. The objective of MPP is to predict the position of the missing part. The ability to solve this task indicates that computers can identify flaws in a story.

In this study, we summarize our propositions and findings in [MYMH20]. Then, we conduct further analysis on the results we obtained in the previous research.

In: R. Campos, A. Jorge, A. Jatowt, S. Bhatia, M. Finlayson (eds.): Proceedings of the Text2Story'21 Workshop, Online, 1-April-2021, published at http://ceur-ws.org

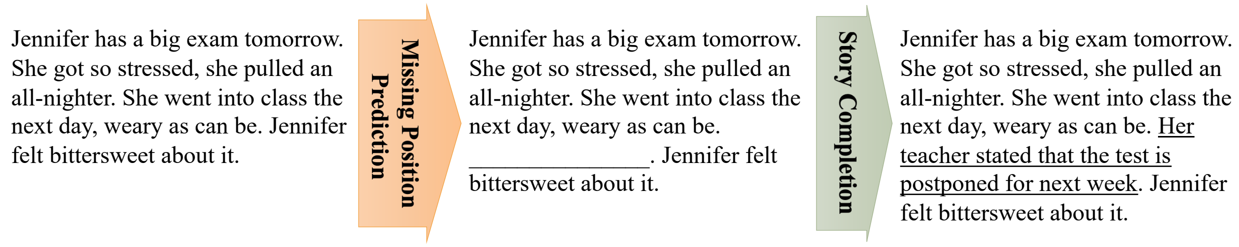

Figure 1: Example of an Incomplete Story and Flow of MPP and SC

2 Proposed Task and Method

2.1 Missing Position Prediction

We define $S = \{s_1, s_2, ..., s_n\}$ as a story comprising n sentences. $s_i (i = 1, 2, ..., n)$ represents a sentence. An input for MPP is an incomplete story comprising $n-1$ sentences $S' = \{s_1, ..., s_{k-1}, s_{k+1}, ..., s_n\}$, where one sentence s_k is dropped from the complete story S. The vital thing in this task is that no information about k is provided. Our objective is to predict k from the input, i.e. remaining sentences S'. The model is trained to maximize the probability $p(missing = k|S')$. Note that the order of the sentences is kept. Specifically, s_{k-1} and s_{k+1} are treated as continuous sentences.

2.2 Proposed Method

Hierarchical approaches have demonstrated effectiveness in story generation [FLD18, RWM+18]. Most likely, the reason is that a typical story has a hierarchical structure. Referring to them, we proposed a method with a two-step encoder. The first encoder, called **Sentence Encoder**, receives S' and outputs the sentence embeddings. As the sentence encoder, We apply Sentence-BERT (SBERT) [RG19] in each sentence. Next, the second encoder, called **Context Encoder**, receives the sentence embeddings and generates a distributed representation of the entire context $v_{context}$. We hypothesize that the input should be treated as a sequence because the order of the original sentences is preserved. Our experiment showed that a gated recurrent unit (GRU) [CvMG+14] is useful as the main part of the context encoder. The output of the GRU is input into a linear layer and a batch normalization layer [IS15]. Then, we input $v_{context}$ into a linear layer and obtain a five-unit output to obtain the MPP result.

Although the stories we used for experiments in [MYMH20] are limited to five-sentence short stories, a story with five sentences is sufficiently long to have minimal context [MCH+16]. Furthermore, the task design of MPP is not limited to the case of $n = 5$ and can be extended for even longer stories as well. Hence, we considered a hierarchical approach is suitable for performing this task.

3 Result of Experiment and Further Analysis

Table 1: Overview of the Dataset Used

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

3.1 Dataset

For our experiments, we used the ROCStories dataset [MCH+16], a collection of 98,161 non-fictional daily-life stories. As shown in Table 1, we split the dataset in a ratio of 8:1:1 and created train/validation/test sets. For each story, one sentence was randomly excluded to create an incomplete story S'. The missing position k was randomly determined based on a discrete uniform distribution. 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

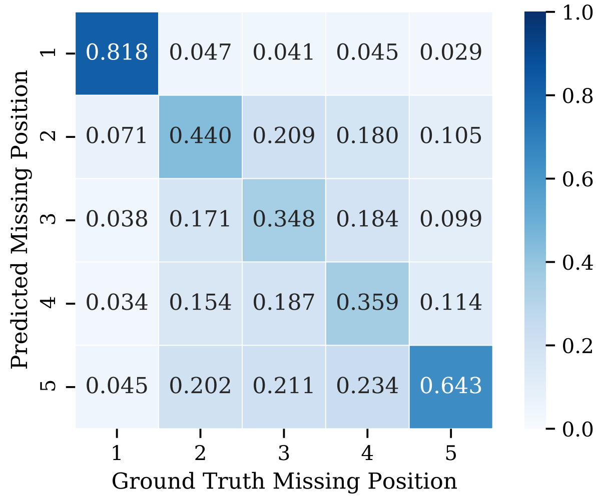

Figure 2: Prediction Accuracy

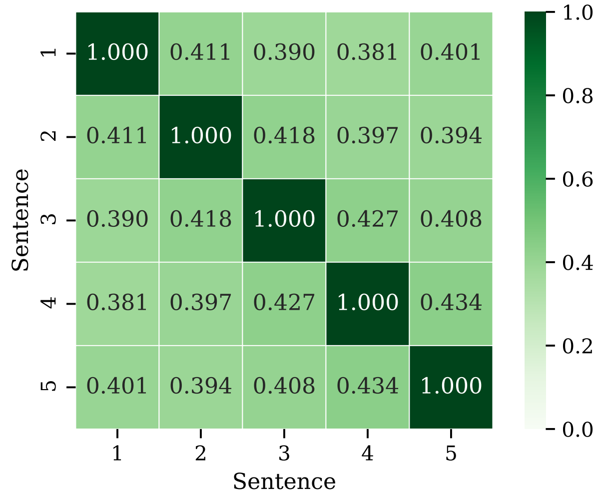

Figure 3: Cosine Similarity Matrix

a result, a different sentence could be removed from the same story, with a different k value, thus acting as data augmentation. For the validation (development) set and the test set, the removal procedure was performed when creating the dataset to improve reproducibility.

3.2 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," where the output dimension was 768. The Context Encoder consists of a GRU with 256 hidden units, and a linear layer with 256 dimensions for both the input and output. The weights of the linear layer were initialized from a normal distribution with $mean = 0$ and $std = 0.01$. To obtain the five-class prediction, we use another linear layer to receive the Context Encoder's output with 256 dimensions and subsequently outputs five dimensions. 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. Gradient clipping with a value of five was used. The batch size was set to 256.

3.3 Result and Further Analysis

The prediction accuracy of each position is shown in Fig. 2 When the first or fifth sentence was missing, the accuracy was higher than when the second, third, or fourth sentence was missing. In other words, the beginning or the ending of a story can be easily predicted when they are the lost sentence. This appears to be related to how ROCStories was collected: "the story should read like a coherent story, with a specific beginning and ending, where something happens in between." [MCH+16] Thus, it is likely that if the beginning or the ending is missing, our method can interpret it as unnatural.

In this study, we analyze this fact further. First, we hypothesize that the second to fourth sentences in a five-sentence story are represented by a similar sentence embedding. For each story in the training set, we considered the sentence embeddings of five sentences and calculated the cosine similarity between them. Fig. 3 shows the average cosine similarity matrix of all stories in the train set. Contrary to our hypothesis, the sentence embeddings of the second to fourth sentences were not more similar to each other than to the first and fifth sentences.

Next, we conducted an analysis using part-of-speech (POS) tagging. POS tagging of words in each sentence was performed for all stories in the training set. For each sentence number, we took the average number of times each tag appeared in that sentence. Further, we used spaCy [HMVLB20] to obtain the universal POS tags. Fig. 4 shows the result. The focus is on whether there is anything special about the distribution of POS tags in the first and last sentences. In the first sentence, the appearance of PROPN (proper noun) is remarkable, and in the fifth sentence, although not as prominent, ADV (adverb) and ADJ (adjective) occur frequently. Whether the prediction models actually pay attention to words with these tags is a subject for future analysis.

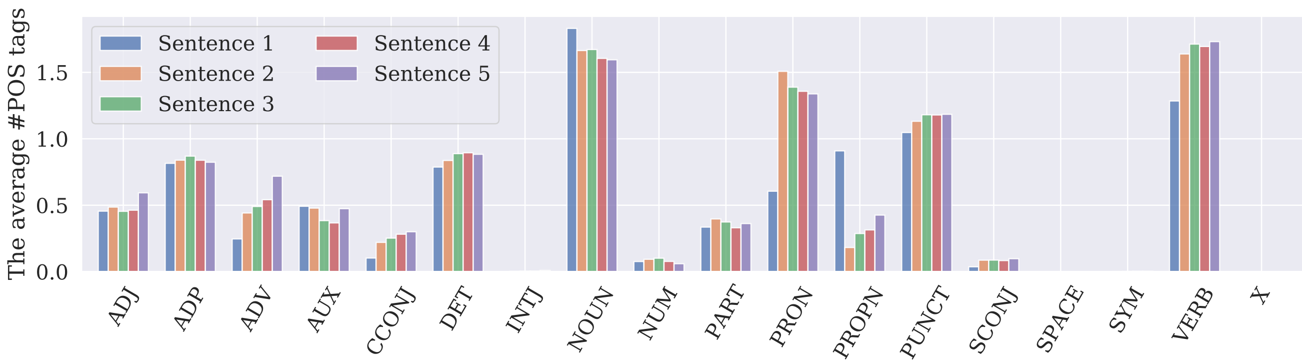

Figure 4: The average #POS tags appear in each sentence in five-sentence stories.

4 Conclusion

To overcome the limitation of the conventional story completion task, we proposed "Missing Position Prediction" to predict the position of the missing part based on the given incomplete story [MYMH20]. We examined the prediction accuracy of our proposed method and found that a prediction is easier if the beginning or the end of a story is missing.

In this study, we summarized what we had proposed and found in [MYMH20] and conducted the further analysis. The analysis suggested that the distribution of part-of-speech tags may play a significant role in prediction accuracy.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP19H01115, JP20H05556, and Moonshot R&D Grant Number JPMJPS2011. We would like to thank anonymous reviewers for their helpful suggestions and comments.

References

[CvMG+14] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 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 2014. Association for Computational Linguistics.

[FLD18] Angela Fan, Mike Lewis, and Yann Dauphin. 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 2018. Association for Computational Linguistics.

[GTFP19] Seraphina Goldfarb-Tarrant, Haining Feng, and Nanyun Peng. 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 2019. Association for Computational Linguistics.

[HMVLB20] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[IS15] Sergey Ioffe and Christian Szegedy. 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 2015. PMLR.

[MCH+16] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 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 2016. Association for Computational Linguistics.

[MYMH20] Yusuke Mori, Hiroaki Yamane, Yusuke Mukuta, and Tatsuya Harada. Finding and generating a missing part for story completion. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 156–166, Online, December 2020. International Committee on Computational Linguistics.

[PGMK18] Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[RG19] Nils Reimers and Iryna Gurevych. 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 2019. Association for Computational Linguistics.

[Roe16] Melissa Roemmele. 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 2016. AAAI Press.

[RWM+18] Hareesh Ravi, Lezi Wang, Carlos Muniz, Leonid Sigal, Dimitris Metaxas, and Mubbasir Kapadia. Show me a story: Towards coherent neural story illustration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[WW19] Tianming Wang and Xiaojun Wan. 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 2019.

[YPR+19] Lili Yao, Nanyun Peng, Weischedel Ralph, Kevin Knight, Dongyan Zhao, and Rui Yan. 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 2019. AAAI Press.