TMU Feedback Comment Generation System Using Pretrained Sequence-to-Sequence Language Models

Naoya Ueda and **Mamoru Komachi**
Tokyo Metropolitan University
ueda-naoya@ed.tmu.ac.jp, komachi@tmu.ac.jp

Abstract

In this paper, we introduce our Tokyo Metropolitan University Feedback Comment Generation system submitted to the feedback comment generation task for INLG 2023 Generation Challenge. In this task, a source sentence and offset range of preposition uses are given as the input. Then, a system generates hints or explanatory notes about preposition uses as the output. To tackle this generation task, we finetuned pretrained sequence-to-sequence language models. The models using BART and T5 showed significant improvement in BLEU score, demonstrating the effectiveness of the pretrained sequence-to-sequence language models in this task. We found that using part-of-speech tag information as an auxiliary input improves the generation quality of feedback comments. Furthermore, we adopt a simple postprocessing method that can enhance the reliability of the generation. As a result, our system achieved the F1 score of 47.4 points in BLEU-based evaluation and 60.9 points in manual evaluation, which ranked second and third on the leaderboard. [1].

1 Introduction

This paper describes our submission to the feedback comment generation task for INLG 2023 Generation Challenge (Nagata et al., 2021). Feedback comment generation is a task of automatically generating hints or explanatory notes about errors for the purpose of helping the language learner improve their writing skills (Nagata, 2019). In this task, the target of the feedback comment generation is limited to preposition uses, such as missing prepositions, to-infinitives, and deverbal prepositions. Table 1 shows the overview of this task.

In the previous study (Hanawa et al., 2021), Pointer Generator Network (See et al., 2017) was used as a sequence-to-sequence method and found to be effective in a setting with few variations of feedback comments such as preposition uses. While this study shows the effectiveness of non-pretrained sequence-to-sequence models such as Pointer Generator Network, no experiments using pretrained language models have been conducted. Since pretrained sequence-to-sequence language models, such as T5 (Raffel et al., 2020), show significant performance in the generation task, it is conceivable that using pretrained sequence-to-sequence language models improves the generation quality.

Input

Source sentence: I can not agree you in this case.
Offeset Ranges: 9:18

Output

Since the <verb> «agree» is an <intransitive verb>, a <preposition> needs to precede the <object>. Look up the <verb> «agree» in the dictionary to find the appropriate <preposition>.

Figure 1: Overview of the feedback comment generation task.

In this paper, we examined the performance of pretrained sequence-to-sequence language models in the feedback comment generation task. We employ BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) as the pretrained sequence-to-sequence language models. Both models have improved the generation quality compared with the non-pretrained sequence-to-sequence model. Also, we confirmed that using part-of-speech (POS) tags as an auxiliary input improves the generation quality of feedback comments in the T5 model. Furthermore, we adopted a simple postprocessing method

[1] Our source code is available at https://github.com/NOIRUED/T5_FCG.git

Proceedings of the 16th International Conference on Natural Language Generation: Generation Challenges, pages 68–73

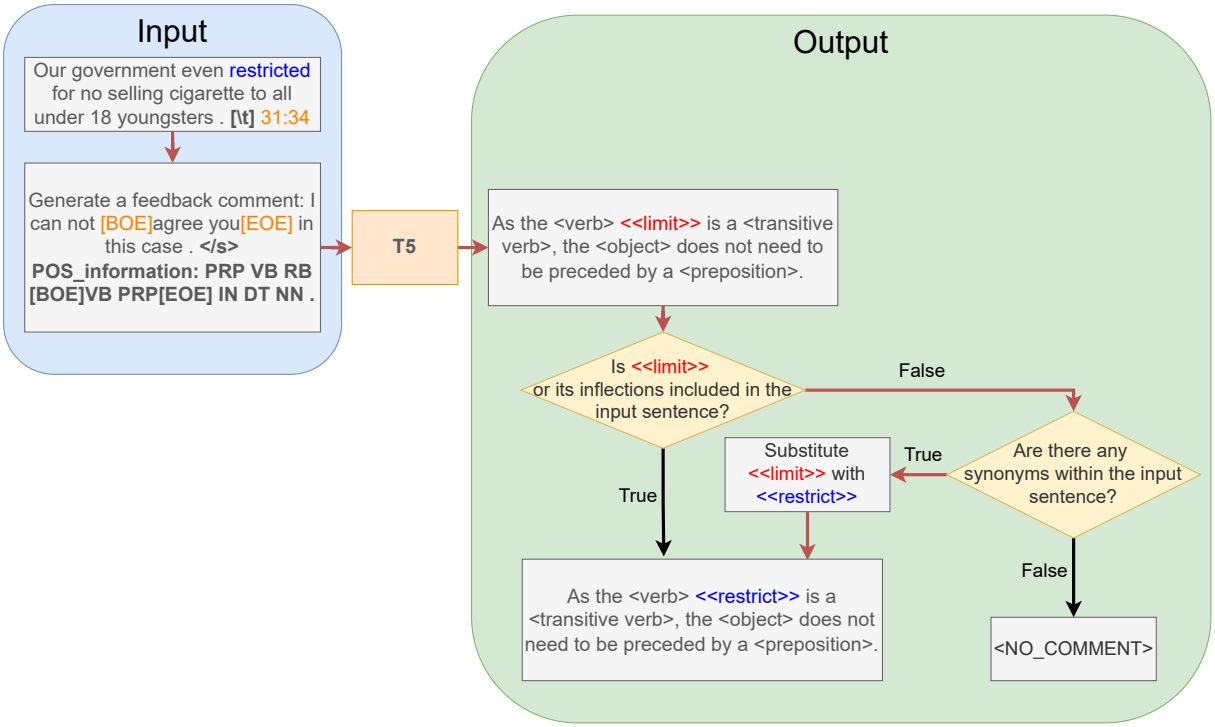

Figure 2: Overview of our method.

to enhance the reliability of the generation. By using this model and methods, we achieved the F1 score of 47.4 points in the BLEU-based evaluation and 60.9 points in the manual evaluation.

2 Feedback Comment Generation Task

2.1 Task Description

The task focuses on the feedback comment generation targeted on preposition uses. As the input, the source sentence and the offset ranges indicating where to comment is given. From the input, a system is required to generate an appropriate feedback comment or the special token <NO_COMMENT> indicating that the system cannot generate any reliable feedback comment.

2.2 Evaluation

The performance of the system is evaluated automatically and manually. As an automatic evaluation, BLEU (Papineni et al., 2002) score is calculated between the system output and the reference using SacreBLEU (Post, 2018). A manual evaluation is done by the shared task organizers on the final submission. Both evaluations are measured by recall, precision, and F1. System outputs with <NO_COMMENT> are excluded from both the numerator and the denominator of precision and the numerator of recall.

2.3 Official Baseline System

The official baseline system is Pointer Generator Network model (See et al., 2017) implemented based on fairseq (Ott et al., 2019). It is a sequence-to-sequence neural network with attention and copy mechanisms. We refer to this model as a non-pretrained sequence-to-sequence model and compare it with pretrained sequence-to-sequence models.

3 Our Method

We frame the feedback comment generation task as a sequence-to-sequence generation task. We finetuned the pretrained sequence-to-sequence language models with the official distributed datasets. Since it is difficult for the models to learn the meaning of the offset ranges, instead of using offset as it is, we inserted the special tokens [BOE] and [EOE] in the position of offset ranges. Figure 2 shows the overview of our proposed method.

3.1 Auxiliary Input

As shown in Figure 2, there are cases that POS information is needed in the output. However, the input sequence does not contain such information, which might lead a system to generate a feedback comment with wrong POS information.

	Train	Dev	Test
Official Datasets	4,868	170	215

Table 1: Number of data instances used in the experiment.

To address this problem, we used POS tag information as an auxiliary input in the T5 model. We used Natural Language Toolkit (NLTK) (Bird et al., 2009) to obtain POS tags of the source sentence. Using the obtained POS tags, we concatenated them with the source sentence as follows:

[Source sentence] <\s> POS: [POS tags]

where <\s> is special token in T5. This method (we will refer to as POSTAG hereafter) allows the T5 model to learn the POS information of the source sentence, which makes better auxiliary inputs.

3.2 Postprocessing

In this task, the quotations from the source sentence should be bracketed using double-angle brackets. Conversely, if the double-angle bracketed words are not present in the source text, the feedback comment is considered unreliable. However, there are cases where the T5 model quotes the words that do not exist in the source sentence. To overcome this problem, we adopted a simple postprocessing method (we will refer to it as EDIT hereafter). In this postprocessing method, if the double-bracketed words do not exist in the source sentence, it finds the 10-best synonyms using FastText (Bojanowski et al., 2017). If any of the 10-best synonyms are included in the text, the system replaces the bracketed word with the synonym. Conversely, if none of the 10-best synonyms are included in the text, it changes the outputs to <NO_COMMENT>.

4 Experimental Settings

4.1 Dataset

In this paper, we only used the official datasets distributed in the shared task. Since there are some typographical errors and orthographic variants in the datasets, we preprocessed the datasets to correct typographical errors and unify orthographic variants. The number of data instances is shown in Table 1.

4.2 Model

In this study, we employ BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) as the pretrained sequence-to-sequence language models. We used the Huggingface Transformer (Wolf et al., 2020) to implement the models.

BART For the BART-based model, we use the BART-base [2] and BART-large [3]. For fine-tuning, the models are optimized using AdamW optimizer with the constant learning rate of 1e-5, the batch size 16, and trained for 20 epochs.

T5 For the T5-based model, we use the T5-base [4] and T5-large [5]. For fine-tuning, the models are optimized using AdamW optimizer with a constant learning rate of 5e-4, a batch size of 16, and trained for 30 epochs. To specify a task, the prefix "Generate a feedback comment: " is added at the beginning of input sequences.

System	BLEU		
	Precision	Recall	F1
Official Baseline	46.3	46.3	46.3
BART-base	51.9	51.9	51.9
BART-large	51.6	51.6	51.6
T5-base	**64.0**	**64.0**	**64.0**
T5-large	60.4	60.4	60.4

Table 2: Experimental results for each system.

System	BLEU		
	Precision	Recall	F1
T5-base	64.0	64.0	64.0
+POSTAG	64.7	**64.7**	**64.7**
+EDIT	**64.9**	64.4	64.6

Table 3: Experimental results for POSTAG and EDIT settings.

5 Results

5.1 Exeperimental Results

Table 2 shows the experimental results against the development set. Compared with the official baseline system, BART and T5 models improved the BLEU scores, demonstrating the effectiveness of the pretrained sequence-to-sequence language models in this task. In our case, the T5-base model

[2] https://huggingface.co/facebook/bart-base
[3] https://huggingface.co/facebook/bart-large
[4] https://huggingface.co/t5-base
[5] https://huggingface.co/t5-large

Source sentence	But smoking in the restaurant will cause both the smokers and surrounding people facing with the those problems more than public places .	
System	System Output	BLEU
Gold	A <verb> part representing the cause of <verb> «cause» takes the form of a <to-infinitive> rather than the <ing-form>.	100.00
T5-base	A <verb> part representing the cause of <verb> «cause» takes the form of a <to-infinitive> rather than the <base form>.	89.53
POSTAG	A <verb> part representing the cause of <verb> «cause» takes the form of a <to-infinitive> rather than the <ing-form>.	100.00

Table 4: Example of the result in POSTAG setting. The underline indicates the offset ranges.

Source sentence	With the development of society , we , college students , should do more to adjust it .	
System	System Output	BLEU
Gold	As the <verb> «adjust» is an <intransitive verb> when used to express “to adapt to something” , [...]	100.00
POSTAG	The <verb> «adapt» does not take an <indirect object> to indicate what one adjusts to. Use the <verb> «adapt» as an <intransitive verb> with a <preposition>. [...]	37.72
EDIT	The <verb> «adjust» does not take an <indirect object> to indicate what one adjusts to. Use the <verb> «adjust» as an <intransitive verb> with a <preposition>. [...]	44.97

Table 5: Example of the result in EDIT setting. The underline indicates the offset ranges.

performed best in this task. We expected large-sized models to perform better than the base-sized models, but contrary to our expectations, the base-sized models outperformed the large-sized models. We consider this odd finding comes from a lack of sufficient parallel data or unreliableness of BLEU scores in the feedback comment generation task. We leave for future work a more detailed examination of these model differences.

Table 3 shows the experimental results in the POSTAG and EDIT settings. Compared with the T5-base model, POSTAG setting improved the score by 0.7 points. The improvements of the BLEU score are relatively small because the superficial differences in the generated outputs were small. Table 4 shows the example that the model has successfully used POS tag information. From the table, we can confirm that POSTAG setting generated feedback comments with correct POS information, but the BLEU score only improved by 10.5 points. These results indicate that using POS tag information as an auxiliary input does not improve the overall BLEU score, but is effective in this task to generate reliable feedback comments.

Compared with the POSTAG setting, EDIT setting improved the precision, but lowered recall and F1 score. Although, the EDIT setting does not improve the BLEU score, it actually enhances the reliability of the feedback comments. Table 5 shows the example that had successfully edited an unreliable feedback comment into a reliable feedback comment. These results show that our postprocessing method is effective to enhance the reliability of the generation.

5.2 Official Results

From the experimental results, we submitted the T5-base with POSTAG and EDIT as our final submission to the shared task. As shown in Table 6, our system obtained a BLEU score of 47.4 and a manual evaluation score of 60.9, which ranked second and third on the leaderboard.

System	BLEU			Manual Evaluation		
	Precision	Recall	F1	Precision	Recall	F1
Official Baseline	33.4	33.4	33.4	31.2	31.2	31.2
Our System	**47.7**	**47.1**	**47.4**	**61.3**	**60.5**	**60.9**

Table 6: Official results.

6 Conclusion

In this paper, we described our submission to the feedback comment generation task for INLG 2023 Generation Challenge. The result of the experiments showed that using pretrained sequence-to-sequence language models is effective in the feedback comment generation for preposition uses. Furthermore, we found that using POS tags as an auxiliary input improves the generation quality, and confirmed that our postprocessing method enhances the quality of the feedback comments by editing unreliable feedback comments into reliable feedback comments. Future work will explore additional postprocessing methods that can better identify and appropriately edit unreliable feedback comments.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kazuaki Hanawa, Ryo Nagata, and Kentaro Inui. 2021. Exploring methods for generating feedback comments for writing learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9719–9730, Online and Punta Cana, Dominican Republic. 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. Association for Computational Linguistics.

Ryo Nagata. 2019. Toward a task of feedback comment generation for writing learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3206–3215, Hong Kong, China. Association for Computational Linguistics.

Ryo Nagata, Masato Hagiwara, Kazuaki Hanawa, Masato Mita, Artem Chernodub, and Olena Nahorna. 2021. Shared task on feedback comment generation for language learners. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 320–324, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen,

Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.