The Tokyo Tech and AIST System at the GenChal 2022 Shared Task on Feedback Comment Generation

Shota Koyama[1,2], Hiroya Takamura[2], Naoaki Okazaki[1,2]

[1]Tokyo Institute of Technology
[2]National Institute of Advanced Industrial Science and Technology
shota.koyama@nlp.c.titech.ac.jp
takamura.hiroya@aist.go.jp
okazaki@c.titech.ac.jp

Abstract

This paper describes the Tokyo Tech and AIST system in the GenChal 2022 shared task, which is the first shared task of feedback comment generation. We adopted five methods: data cleaning, fine-tuning pre-trained models, correcting errors in learners' sentences, appending a correcting operation, and filtering out irrelevant outputs. Our system achieved $F_1 = 43.4$ on the test dataset.

1 Introduction

Recently, Nagata (2019) proposed a novel task called feedback comment generation (FCG), wherein feedback is provided to help writers improve their skills, especially in the context of computer-assisted language learning. The input of an FCG model is a learner's sentence, and the output is a comment given as feedback to the learner.

The GenChal 2022 shared task is the first shared task of the FCG task. This paper describes the system developed in our study, which encompasses the following five methods:

(1) **Data cleaning** (§3.1): We corrected mistakes in the annotations within the training dataset.

(2) **Fine-tuning pre-trained models** (§3.2): We fine-tuned pre-trained models to address the low-resource aspect of this task.

(3) **Correcting errors in learners' sentences** (§3.3): We corrected errors in the input sentences outside of the target words for the FCG, thus preventing errors in the model output.

(4) **Appending a correcting operation** (§3.4): We appended a correcting operation (such as "delete") to the input with the aim of generating more accurate feedback comments.

(5) **Filtering** (§3.5): We removed irrelevant feedback comments using simple heuristics.

This paper is organized as follows. Section 2 describes the shared task and its dataset. Section 3 details the methods outlined above. Section 4 presents the experimental setup. Section 5 shows the results. Section 6 concludes this paper.

All of our code has been publicly released for reproducibility[1].

2 Task and Dataset Description

The GenChal 2022 shared task was proposed by Nagata et al. (2021) to address FCG. The organizers released a new dataset for this task, wherein original texts written by English learners were borrowed from ICNALE (Ishikawa, 2011).

The input of this task is a pair consisting of the learners' text and a span indicating the feedback comment's location. The input text is written in English and tokenized. The span is provided as input in this task, although it can be detected by grammatical error detection models. For example, the sentence "`It is a problem for health .`" has an error, which we can correct by replacing `for` to `of`. The span is character-level and colon-separated, and the position indicates a 0-indexed point between characters including whitespace. Therefore, the span of this example is `16:19`, wherein the start position is 16 and the end position is 19.

The output of this task is feedback comment, which must be informative beyond merely an indication of 'correct' or 'incorrect'. Furthermore, specific words and phrases in feedback comments are annotated using brackets. Grammar terms and idiomatic patterns are bracketed using < >, whereas quotations from the learner's sentence are bracketed using << >>. Miscellaneous quotations and words or phrases to highlight can be annotated using `‘‘ ''`. For example, the feedback comment for the sentence in the last paragraph can be: `The <preposition> <<for>> should precede a person. Simply use ‘‘of'' in this case.` The special output `<NO_COMMENT>` indicates that the system cannot generate any reliable feedback

[1]https://github.com/shotakoyama/fcgtools

Proceedings of the 16th International Natural Language Generation Conference: Generation Challenges, pages 74–78
September 11–15, 2023.

comment. In this task, feedback comments are written in English, while another choice is using the learner's native language as in Nagata et al. (2020).

The released dataset was split into training, development, and test subsets, which contain 4868, 170, and 215 sentence pairs, respectively. Feedback comments in the test dataset were not released during the shared task period.

The BLEU (Papineni et al., 2002) score is adopted as the automatic evaluation metric. To incorporate the use of <NO_COMMENT> into the evaluation, task submissions are evaluated by BLEU-based F_1 score. The precision (P), recall (R) and F_1 score are calculated as follows:

$$\mathcal{R} : \text{reference sentences } (r_1, r_2, \cdots),$$
$$\mathcal{S} : \text{system outputs } (s_1, s_2, \cdots),$$
$$\mathcal{S}' = \{s_i \in \mathcal{S} | s_i \neq \texttt{<NO\_COMMENT>}\},$$
$$P = \frac{1}{|\mathcal{S}'|} \sum_{i \in \{i | s_i \in \mathcal{S}'\}} \text{BLEU}(s_i, r_i),$$
$$R = \frac{1}{|\mathcal{R}|} \sum_{i=1}^{|\mathcal{R}|} \text{BLEU}(s_i, r_i),$$
$$F_1 = 2 \times \frac{P \times R}{P + R}.$$

3 Method

3.1 Data Cleaning

Because we found the training data contain many annotation mistakes, we manually cleaned all erroneous feedback comments. 1,770 data samples (≈37%) were affected by this process.

3.1.1 Illegal Span

The start and end of the target's span must correspond to the start and end of a word, respectively. For example, a span of `10:12` is correct for the input "`It is fun to me .`", whereas a span of `9:12` is illegal.

3.1.2 Wrong Annotation

Annotations using brackets must satisfy the bracket correspondence. Illegal brackets (e.g., <verb>> → <verb>) and illegal quotations (e.g., `'of''` → ‘‘`of''`) are corrected.

3.1.3 Others

Some trivial mistakes include grammatical errors and the usage of non-ASCII characters. Please refer to the source code for all modifications[2].

3.2 Fine-Tuning Pre-Trained Models

Recently, many NLP studies have focused on the use of pre-trained models that are trained on unlabeled data. Pre-training and subsequently fine-tuning is a simple and effective approach widely adopted for low-resource NLP tasks. We fine-tuned a Transformer decoder model, GPT-2 (Radford et al., 2019), and an encoder-decoder model, BART (Lewis et al., 2020).

Because the model requires access to the target position in the input, we added double brackets to the target of feedback comment generation. For example, if the input is "`I agree the issue .`" and the span is `2:11`[3], the model input is "`I <<agree the>> issue .`".

Figure 1 illustrates the use of the BART and GPT-2 models for this task. Because the learner's sentence and feedback comment must be unified as the input of the GPT-2 decoder, the two sentences are concatenated with #[4]. In training, the GPT-2 model predicts entire concatenated sentences[5].

3.3 Correcting Errors in Learners' Sentences

Learners' sentences may have many errors outside of the target range, which can negatively impact performance. For example, the input "`I want go <<to>> abroad .`" has a non-targeted error and would be modified to "`I want` **`to`** `go <<to>> abroad .`". To address this issue, we corrected any non-targeted errors using GECToR (Omelianchuk et al., 2020), one of the state-of-the-art grammatical error correction models.

3.4 Appending a Correcting Operation

The GECToR model corrects errors by predicting correcting operations, such as "delete" and "from base form to -ing form", which can benefit more accurate feedback comment generation. We extracted the GECToR tag for the input sentence's target word, replaced it with a more intuitive form

[2] https://github.com/shotakoyama/fcgtools/blob/main/fcgtools/cli/prepare.py

[3] This span indicates that this correction inserts a word (in this case on) between agree and the.

[4] We tried various other separation tokens and found that the token selection is not significant with regards to the performance.

[5] In our preliminary experiments, conducting back-propagation of both learner's sentence and feedback comment yielded better performance than that of only feedback comment.

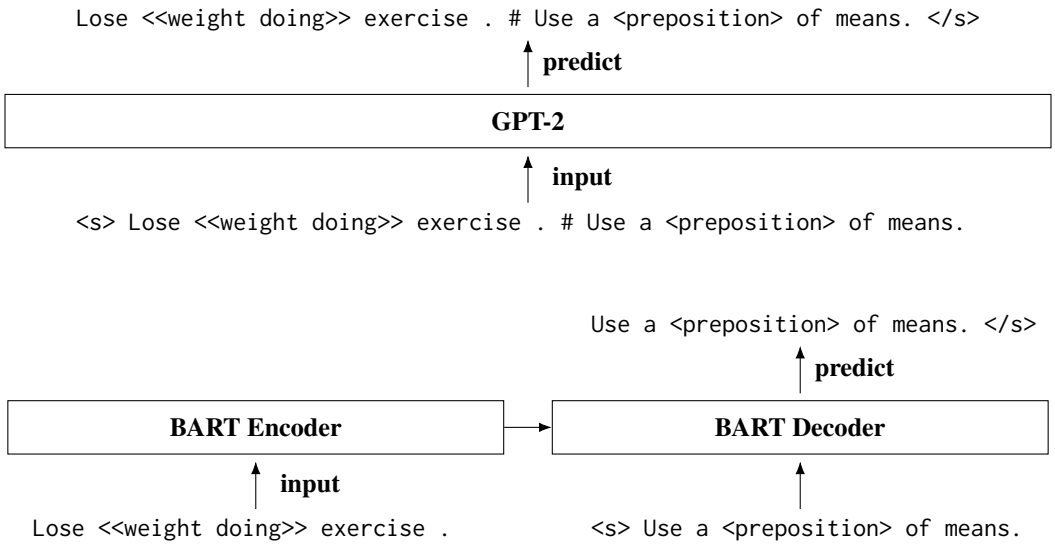

Figure 1: GPT-2 and BART models for the feedback comment generation task.

GECToR tag	tag
DELETE	delete
REPLACE_*	replace *
APPEND_*	append *
TRANSFORM_AGREEMENT_PLURAL	plural
TRANSFORM_AGREEMENT_SINGULAR	singular
TRANSFORM_CASE_CAPITAL	titlecase
TRANSFORM_CASE_CAPITAL_1	capitalcase
TRANSFORM_CASE_LOWER	lowercase
TRANSFORM_CASE_UPPER	uppercase
TRANSFORM_SPLIT_HYPHEN	split hyphen
TRANSFORM_VERB_*_*	from * to *

Table 1: Replacement rules of GECToR tag.

according to the rules listed in Table 1, and concatenated it with `//`. For example, the GECToR tag for the input "`You cannot stop <<to smoke>> .`" should be "`TRANSFORM_VERB_VB_VBG`", and the input would be converted to "`You cannot stop <<to smoke>> . // from VB to VBG`".

3.5 Filtering Out Irrelevant Outputs

Filtering represents a simple heuristic to improve performance. If the quoted part in the feedback comment does not appear in the input sentence, the whole output is replaced with `<NO_COMMENT>`, as the comment is obviously irrelevant. This procedure was introduced to prevent the score from dropping. For example, when the model generated the feedback comment "`Since <<ahead>> is an <adverb>, ...`" for the input "`I want to go <<to>> abroad .`", this comment was filtered out and replaced with `<NO_COMMENT>` because ahead does not appear in the input sentence.

4 Experimental Setup

We used the dataset released by the shared task and selected the best epoch for each training trial using the validation dataset. We fine-tuned models for 100 epochs with saving checkpoints at five-epoch intervals.

We used the cross-entropy loss, AdamW optimizer (Loshchilov and Hutter, 2019) with a weight decay of 0.01 and gradient clipping of 1.0, and a learning rate of 0.0001 with a constant scheduler.

We employed BART small and GPT-2 small to compare both models and GPT-2 large for the final submission. We used NVIDIA A100 GPU with 40 GiB memory for all experiments and varied the batch size for model size to ensure an efficient use of the GPU memory. We set the maximum tokens per batch to 2,000 for BART/GPT-2 small and 250 for GPT-2 large, and accumulated every four batches for BART/GPT-2 small and 32 for GPT-2 large, thus setting the number of maximum tokens for each step to 8,000.

5 Experimental Results

5.1 Comparison Between BART and GPT-2

First, we conduct experiments to compare the performance of BART and GPT-2 and verify the effectiveness of the methods introduced in Section 3.

Table 2 lists the average scores on the validation set obtained by the five models. Correcting non-target errors (+ correction, § 3.3), appending

	BART small	GPT-2 small
fine-tuning	47.74	49.45
+ correction	47.36	50.23
+ operation	47.58	50.70
+ both	47.32	**51.80**

Table 2: Comparison between BART and GPT-2.

	w/o filtering	w/ filtering
fine-tuning	49.45	49.97
+ correction	50.23	50.71
+ operation	50.70	51.14
+ both	51.80	**52.44**

Table 3: Effect of filtering.

a correcting operation (+ operation, § 3.4), and applying both methods improves the performance of GPT-2 and decreases that of BART. Furthermore, GPT-2 performs better than BART in all settings. Accordingly, we selected GPT-2 for the task submission.

5.2 Impact of Filtering

We verified the effectiveness of filtering (§ 3.5) on GPT-2 small. Table 3 lists the average scores on the validation set obtained by the five models. We confirmed that filtering improves the performance by approximately 0.5 points in every setting.

5.3 Final Submission

We compared the results obtained by GPT-2 small and large, to determine the final submission. The results listed in Table 4 represent the best scores on the validation set obtained by the five models. We adopted GPT-2 large, appending a correcting operation and filtering for the final submission. In the shared task, our final submission achieved 43.4 in F_1 score on the blind test set.

6 Conclusion

In this paper, we described our system for the GenChal 2022 shared task. We employed five methods: data cleaning, fine-tuning pre-trained models, correcting errors in learners' sentences, appending a correcting operation, and filtering. We fine-tuned BART and GPT-2 and then selected GPT-2 for submission. We verified that filtering <NO_COMMENT> using a simple heuristic improves performance. Our final submission was obtained using GPT-2 large with appending a correcting operation and filtering without correcting non-target errors. Our system achieved an F_1 score of 54.73 on the validation set, and 43.4 on the test set.

correction	operation	filtering	small	large
✓	✓		52.84	53.96
✓		✓	52.56	52.15
	✓	✓	51.79	**54.73**
✓	✓	✓	53.19	54.33

Table 4: Comparison between GPT-2 small and large.

7 Acknowledgments

This paper is based on results obtained from a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). For experiments, computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used.

References

Shin'ichiro Ishikawa. 2011. A new horizon in learner corpus studies: The aim of the ICNALE project. *Corpora and Language Technologies in Teaching, Learning and Research*, pages 3–11.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

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.

Ryo Nagata, Kentaro Inui, and Shin'ichiro Ishikawa. 2020. Creating corpora for research in feedback comment generation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 340–345, Marseille, France. European Language Resources Association.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. 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.

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