Corrections Meet Explanations: A Unified Framework for Explainable Grammatical Error Correction

Anonymous ACL submission

Abstract

Grammatical Error Correction (GEC) faces the important yet challenging issue of explainability, especially when GEC systems are developed for language learners who often struggle to understand the correction results without reasonable explanations. Extractive evidence words and grammatical error types are two crucial factors of GEC explanations. However, existing work focuses on extracting evidence words and predicting grammatical error types given a source sentence and/or a target sentence as input, ignoring the interaction between explanations and corrections. To bridge the gap, we introduce **EXGEC**, a unified explainable GEC framework that jointly perform explanation and correction tasks in a sequence-to-sequence generation manner, hypothesizing both tasks would benefit each other. Extensive experiments enable us to fully understand and establish the interaction between tasks. Especially, if models are required to jointly predict corrections and explanations, the performance of both tasks improves compared to their respective single-task baselines. Additionally, we observe that EXPECT, a recent explainable GEC dataset, contains considerable noise that may confuse model training and evaluation. Therefore, we rebuild EXPECT to eliminate the noise, leading to an objective training and evaluation pipeline [1].

1 Introduction

Writing is a learnt skill that is particularly challenging for second-language (L2) speakers, who often struggle to create grammatical and comprehensible texts (Bryant et al., 2022). To address the problem of ungrammatical writing, GEC systems are designed to identify and correct all grammatical errors in texts. Research in the field of GEC has extended to include multi-language (Rothe et al., 2021), multi-modality (Fang et al., 2023), document-level (Yuan and Bryant, 2021) and domain adaptation (Zhang et al., 2023).

However, the explainability of GEC is still underdeveloped due to its inherent challenges (Hanawa et al., 2021; Kaneko et al., 2022). Since neural GEC systems are typically complex black-box systems, their inner working mechanisms are opaque (Zhao et al., 2023). The lack of explainability can lead to insufficiency in an educational context, where L2-speakers may struggle to thoroughly grasp the writing skills from GEC systems without understanding why a correction is needed. Equipping corrections with explanations builds appropriate trust by elucidating the linguistic knowledge and reasoning mechanism behind model predictions in an understandable manner, assisting pedagogically end users with elementary language proficiency (Bitchener et al., 2005; Sheen, 2007). Additionally, explainability provides insight to identify unintended biases and risks for researchers and developers, acting as a debugging aid to quickly advance model performance (Ludan et al., 2023).

To help language learners better understand why GEC systems make a certain correction, Fei et al. (2023) introduce EXPECT, a large dataset annotated with *evidence words* and *grammatical error types*. Evidence words, which are formally called extractive rationales [2], provides specific clues for corrections, helping L2-speakers understand "why to correct". the error types in EXPECT cover 15 pragmatism-based categories (Skehan, 1998; Gui, 2004), facilitating L2-speakers in inferring abstract grammar rules from specific errors in an inductive reasoning manner. However, Fei et al. (2023) focus on explaining GEC given an ungrammatical source and/or a corrected sentence, ignoring the interaction between explanation and correction

[1] All the source codes and data will be released after the review anonymity period.

[2] We use the term "evidence words" throughout the paper except Section 6, following Fei et al. (2023).

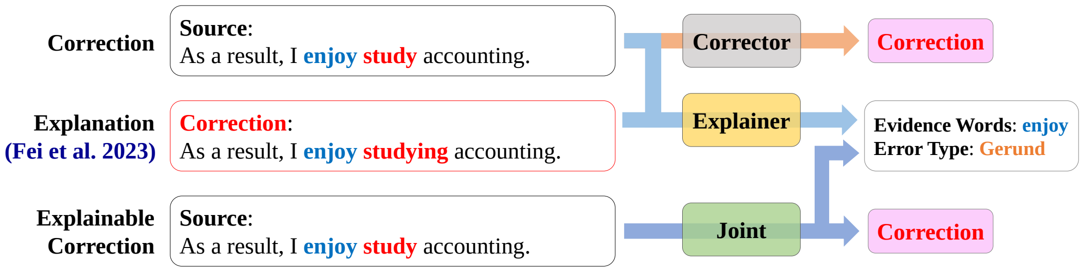

Figure 1: Comparison between correction, explanation (Fei et al., 2023) and our explainable GEC.

tasks, as shown in Figure 1. Previous studies have shown that training models to jointly output task predictions and explanations can improve the task performance on vision-language tasks (Majumder et al., 2022) and diversity downstream NLP tasks, including text classification (Li et al., 2022a), commonsense reasoning (Veerubhotla et al., 2023), and complaint detection (Singh et al., 2023).

To establish the interaction between explanation and correction tasks, we propose **EXGEC** (**EX**plainable **G**rammatical **E**rror **C**orrection), a unified explainable GEC framework that reframes the multi-task problem as a sequence-to-sequence (Seq2Seq) generation task. With pointing mechanism (Vinyals et al., 2015), EXGEC can extract evidence words by directly generating source indexes of an ungrammatical source sentence in an auto-regressive manner. EXGEC can jointly correct ungrammatical sentences, extract evidence words and classify grammatical errors in a unified architecture. To the best of our knowledge, we first propose to jointly perform both correction and explanation tasks. Our findings illustrate that learning correction and explanation tasks concurrently can benefit each other. Specifically, pre-explaining models achieve higher correction performance yet lower explanation performance than post-explaining models. However, both models achieve better or comparable correction and explanation performance than their respective baselines.

Additionally, we observe that EXPECT is not a well-specified dataset for explainable GEC. This is due to the presence of considerable unidentified grammatical errors in EXPECT, which hinder the performance of both tasks. As a result, we rebuild EXPECT to re-correct the unidentified errors while ensuring that each sentence contains only a single unique error, as described by Fei et al. (2023). By training on rebuilt EXPECT, we significantly improve the performance of both tasks, demonstrating the effectiveness of our rebuild process.

2 Rebuilt EXPECT Dataset

In this paper, we utilize the EXPECT dataset (Fei et al., 2023). The dataset comprises a total of 20,016 samples that are split into train, dev and test sets. EXPECT is annotated based on the high-quality GEC dataset, W&I+LOCNESS (Bryant et al., 2019), which is designed to represent a much wider range of English levels and abilities than previous corpora. To reduce the difficulty of the model learning and evaluation, EXPECT is constructed using a special process. Specifically, for a sentence from W&I+LOCNESS with n grammatical errors, the authors repeat the sentence n times and keep a single unique error in each sentence. Considering the challenges of explainable GEC, it is reasonable and desirable as it smooths the task by classifying a grammatical error and extracting evidence words for a single unique grammatical error each time, avoiding the confusion caused by multiple interactive grammatical errors in a sentence.

However, we argue that the official EXPECT dataset is not well-specified. Specifically, for a sentence with $n(n > 1)$ grammatical errors from W&I+LOCNESS, the authors correct a single grammatical error and leave the remaining $n-1$ errors unidentified, as shown in Table 1. These unidentified grammatical errors may confuse models, making it uncertain which error should be corrected and explained, and leading to uncertainty in model training and evaluation. To address the problem, we re-correct the unidentified grammatical errors, while leaving the single original grammatical error unchanged. The entire rebuilding process is automatic since we re-correct all the unidentified grammatical errors by comparing sentences from EXPECT and W&I+LOCNESS. We first retrieve the original parallel samples of W&I+LOCNESS by using the open-source toolkit `TheFuzz` [3], and then identify and correct the un-

[3] https://github.com/seatgeek/thefuzz

W&I+LOCNESS Source	However I sometimes do a skipping to fit myself .
W&I+LOCNESS Target	However , I sometimes do skipping to keep myself fit .
EXPECT Source	However I sometimes do skipping to keep myself .
EXPECT Target	However I sometimes do skipping to keep myself fit .
Rebuilt Source	However , I sometimes do skipping to keep myself .
Rebuilt Target	However , I sometimes do skipping to keep myself fit .
W&I+LOCNESS Source	i have a dog it name 's chente , it is a golden retriver .
W&I+LOCNESS Target	I have a dog and its name 's Chente . It is a golden retriever .
EXPECT Source	i have a dog its name 's chente , it is a golden retriver .
EXPECT Target	i have a dog and its name 's chente , it is a golden retriver .
Rebuilt Source	I have a dog its name 's Chente . It is a golden retriever .
Rebuilt Target	I have a dog and its name 's Chente . It is a golden retriever .

Table 1: Examples of our rebuilt EXPECT. We mark grammatical errors in blue and corrections in red.

		Train	Dev	Test
Official	**#Sent.**	15,187	2,413	2,416
	#Evi. Sent.	11,261	1,426	1,444
	Perc.	74.15%	59.10%	59.77%
	Avg. Words	28,68	29.06	29.23
	Avg. Edits	1.03	1.08	1.07
	Avg. EW/Sent.	2.59	3.00	3.01
Rebuilt	**#Sent.**	15,187	2,413	2,416
	#Evi. Sent.	11,261	1,425	1,443
	Perc.	74.15%	59.06%	59.73%
	Avg. Words	28.52	29.53	29.72
	Avg. Edits	1.03	1.08	1.07
	Avg. EW/Sent.	2.59	3.00	3.00

Table 2: Statistics of the official and rebuilt EXPECT datasets, including the number of sentences (#Sent.), the average number of words per sentence (Avg. Words), the average number of edits per sentence (Avg. Edits), the number and percentage of sentences with annotated evidence (#Evi. Sent. and Perc.), and the average number of evidence words per sentence (Avg. EW/Sent.).

derlying grammatical errors by leveraging GEC evaluation toolkits ERRANT (Bryant et al., 2017). It is worth noting that the evaluation for the official and rebuilt EXPECT datasets are fairly comparable since the grammatical errors and evidence words are retained during the rebuild process, except for a few extreme cases [4]. Totally, 277 (1.82%), 1,311 (54.33%), and 1,323 (54.76%) sentences in our rebuilt train/dev/test sets differ from their original sentences of official EXPECT. Detailed statistics of both EXPECT datasets are listed in Table 2.

3 Methodology

3.1 Problem Definition

The goal of this work is to perform both correction and explanation tasks jointly in a Seq2Seq-based generation approach. Formally, given an ungrammatical source sentence $X = \{x_0, x_1, \cdots, x_n\}$, where n is the length of the source sentence, joint models are designed to learn both correction and explanation tasks. The correction task involves transforming the ungrammatical source into a grammatical target $Y = \{y_0, y_1, \cdots, y_m\}$, where m is the length of the target. The explanation task consists of two sub-tasks: 1) **classifying** grammatical errors, and 2) **extracting** evidence words. The classification task requires joint models to output a grammatical error type label c ($c \in C$), where C is the set of 15 candidate grammatical error type classes defined in EXPECT. And the extraction task requires models to extract evidence words $E(X) = \{e_0, e_1, \cdots, e_k\} \subset X$ that can provide informative and complete clues for corrections.

3.2 Explainable GEC as Generation Task

To investigate the interaction between explanation and correction tasks, we propose four different training settings, as illustrated in Figure 3: 1) no explanations (*Baseline*), which is the conventional setting, 2) explanations as additional input (*Infusion*), 3) explanations as output (*Explanation*), and 4) explanations as additional output (*Self-Rationalization*). To enable all these settings in a single architecture, we propose EXGEC, a unified generative framework for explainable GEC. In the Infusion setting, we introduce a special token "<sep>" to separate the source sentence and the following explanation, which includes evidence words and an error type. In the Explanation setting, the model generates an explanation given only a source sentence. As for the Self-rationalization setting, models are required to output a correction and an explanation separated by the special token "<sep>". The relative positions of corrections and

[4] One sample from the dev set and one sample from the test set are free from evidence words since their evidence words overlap with the unidentified grammatical errors.

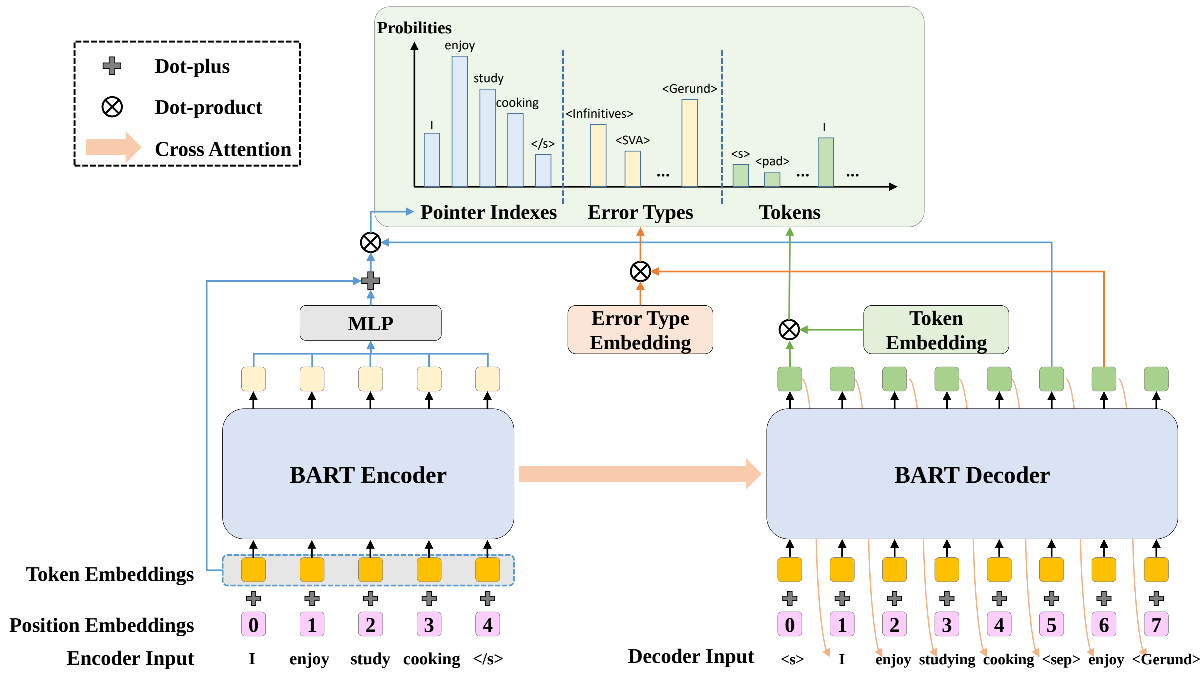

Figure 2: Overview of our Seq2Seq-based *Self-rationalization* model. The decoder can 1) output corrections from BART's token vocabulary, 2) generate evidence words as source indexes by leveraging pointer mechanism, and 3) predict an error type from the predefined set of error type classes.

	Input	Output
Baseline	Source	Correction
Infusion	Source <sep> Evidence Words Error Type	Correction
Explanation	Source	Evidence Words Error Type
Self-rationalization	Source	Correction <sep> Evidence Words Error Type

Figure 3: Comparison of four settings, all of which can be implemented in our proposed unified architecture.

explanations can be reversed, which allows us to understand the interaction between both tasks.

Without loss of generality, we clarify how our EXGEC tackles tasks in a unified generative framework in the Self-rationalization setting. Given an ungrammatical source sentence X, the encoder encodes X into hidden representation $\mathbf{H}$ as follow:

$$\mathbf{H}^e = \text{Encoder}(X), \tag{1}$$

where $\mathbf{H}^e \in \mathbb{R}^{n \times d}$, and d is the hidden size.

At each time step t, the decoder produces the hidden state $\mathbf{h}_t^d \in \mathbb{R}^d$ based on the previous output sequence $\hat{Y}_{<t}$, which is computed as follow:

$$\mathbf{h}_t^d = \text{Decoder}(\mathbf{H}^e, \hat{Y}_{<t}). \tag{2}$$

Next, the hidden state $\mathbf{h}_t^d \in \mathbb{R}^d$ is utilized to calculate three types of logits: 1) *token logits*, which are responsible for the correction part (Vaswani et al., 2017), 2) *pointer logits*, used to determine the probabilities of source indexes for evidence extraction, and 3) *type logits*, utilized for error type classification. Inspired by Yan et al. (2021), we calculate the probability distribution P_t as follows:

$$\mathbf{E}^e = \text{TokenEmbed}(X) \in \mathbb{R}^{n \times d}, \tag{3}$$

$$\bar{\mathbf{H}}^e = \alpha \mathbf{E}^e + (1-\alpha)\,\text{MLP}(\mathbf{H}^e) \in \mathbb{R}^{n \times d}, \tag{4}$$

$$\mathbf{V}^d = \text{TokenEmbed}(V) \in \mathbb{R}^{|V| \times d}, \tag{5}$$

$$\mathbf{C}^d = \text{TypeEmbed}(C) \in \mathbb{R}^{|C| \times d}, \tag{6}$$

$$P_t = \text{softmax}([\mathbf{V}^d \otimes \mathbf{h}_t^d; \bar{\mathbf{H}}^e \otimes \mathbf{h}_t^d; \mathbf{C}^d \otimes \mathbf{h}_t^d]), \tag{7}$$

where TokenEmbed refers to the embeddings that are shared between the encoder and decoder, $\alpha \in \mathbb{R}$ is a hyper-parameter responsible for balancing the trade-off between embeddings and encoder hidden representation, V represents the token vocabulary, $[\cdot\ ;\ \cdot]$ denotes the concatenation operation in the first dimension, the symbol $\otimes$ means the dot product operation, and $P_t \in \mathbb{R}^{|V|+n+|C|}$ represents the probability distribution at the current time step t.

It is worth noting that the pointer index cannot be directly inputted to the decoder, so we introduce the Index2Token conversion to convert indexes into tokens (Yan et al., 2021). Additionally, we can rearrange the generation order of corrections and explanations, which may provide helpful insight into further understanding the interaction of both tasks. In the Baseline and Infusion settings, the probability distribution is limited to the token vocabulary. However, in the Explanation setting, the probability distribution is limited to the combination of pointer indexes and error type classes.

3.3 Loss Weighting

Taking into account the heterogeneity of correction and explanation tasks, we construct the overall loss function in the form of weighted sum, which is defined as follow:

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{cor} + \lambda \cdot \mathcal{L}_{exp} \\ &= -\sum_{i=0}^{m}\Big[\mathbb{I}(y_i \in V)\log p_i + \lambda\mathbb{I}(y_i \notin V)\log p_i\Big],\end{aligned} \tag{8}$$

where λ is responsible for balancing both tasks, and $\mathbb{I}$ is the indicator function. During the inference stage, we generate the entire target sequence in an autoregressive manner and then separate different parts from the target.

4 Experiments

4.1 Experimental Settings

Backbone model. We adopt the Seq2Seq-based pre-trained model BART-Large (Lewis et al., 2020) as our backbone model. All experiments are conducted using the open-source sequence modeling toolkit `Fairseq` (Ott et al., 2019), and subwords are obtained using the byte-pair-encoding (BPE) (Sennrich et al., 2016) algorithm. It is worth noting that adopting BART is non-trivial because the BPE tokenization may split a word into several BPEs, making it tricky to extract evidence words. Considering evidence words are usually short and not always contiguous, we stipulate that the pointer indexes should contain all BPEs of the evidence words. In other words, if a word is an evidence word, models in the Explanation and Self-rationalization settings are desired to output the pointer indexes of all its BPEs. If an instance has no evidence word, the target skips the prediction of pointer indexes. Additionally, we apply the Dropout-Src mechanism (Junczys-Dowmunt et al., 2018) to source-side word embeddings following previous work (Zhang et al., 2022). Detailed hyper-parameter settings are provided in Appendix A.

Training Settings. As discussed in Section 3.2, we attempt to conduct experiments on four distinct training settings leveraging a single unified framework with minimal modification. Notably, the Self-rationalization setting can be further divided into two settings based on the generation order of the correction and explanation parts: 1) *pre-explaining* models first output the explanation part and then the correction part, while 2) *post-explaining* models work in reverse order. In general, we extract evidence words first and then predict error types since we find that the generation order of evidence words and error types does not significantly affect the performance in our preliminary experiments.

Evaluation. We evaluate the model performance in three aspects. 1) Correction. Following the authors of the W&I+LOCNESS dataset (Bryant et al., 2019), we report correction performance evaluated by ERRANT (Bryant et al., 2017). 2) Extraction of evidence words. Following Fei et al. (2023), we also employ token-level evaluation metrics such as Precision, Recall, F_1 and $F_{0.5}$. However, we do *not* adopt the exact match (EM) metric since it is reported to be the least correlated with human evaluation [5]. The findings (Fei et al., 2023) show that the $F_{0.5}$ score achieves the highest correlation with human evaluation in terms of Pearson coefficient, followed by the F_1 score. 3) Classification of grammatical errors. We report label accuracy as the classification performance of grammatical error types. Unlike previous work (Fei et al., 2023), we disentangle the evaluation of extraction and classification, which might provide a clearer perspective on aspects of model performance. Specifically, we deem an evidence word as a True Positive (TP) if all of its BPEs are extracted, which is not in line with the previous evaluation (Fei et al., 2023) that considers an evidence word as a TP only if both BPEs and its error type are correctly predicted. The results are averaged over three runs with different random seeds, and the EXPECT-*dev* set serves as the validation set in all experiments.

4.2 Experiments on Rebuilt Datasets

To demonstrate the effectiveness of our rebuild process, we first respectively train post-explaining

[5] Surprisingly, we find that *do-nothing* systems achieve higher EM scores than almost all well-trained systems, but 0 F_1 and $F_{0.5}$ scores.

System	EXPECT-*dev* Cor. (P / R / $F_{0.5}$)	EXPECT-*dev* Exp. (P / R / F_1 / $F_{0.5}$ / Acc)	EXPECT-*test* Cor. (P / R / $F_{0.5}$)	EXPECT-*test* Exp. (P / R / F_1 / $F_{0.5}$ / Acc)
BART Baseline	36.14 / 34.87 / 35.88	-	36.33 / 35.49 / 36.16	-
BERT Explanation	-	53.60 / 35.46 / **42.68** / **48.63** / **52.09**	-	51.73 / 36.34 / **42.69** / **47.69** / **50.83**
BART Explanation	-	44.43 / 32.93 / 37.82 / 41.53 / 33.36	-	42.34 / 33.13 / 37.18 / 40.11 / 26.95
Infusion				
+ Evidence	45.78 / 44.55 / **45.53**	-	46.02 / 44.13 / **45.63**	-
+ Type	35.31 / 47.87 / 35.22	-	36.00 / 35.37 / 35.87	-
+ Evidence&Type	44.28 / 47.55 / 44.90	-	44.96 / 47.50 / 45.44	-
Self-rationalization				
Pre-explaining	38.25 / 34.18 / **37.36**	36.01 / 35.58 / 35.79 / 35.92 / 26.56	38.68 / 35.41 / **37.98**	36.77 / 36.85 / 36.81 / 36.79 / 26.24
Post-explaining	36.34 / 40.15 / 37.05	48.95 / 42.72 / **45.63** / **47.56** / **40.32**	36.52 / 40.41 / 37.24	49.43 / 44.10 / **46.61** / **48.26** / **39.86**

Table 3: Results of different settings for the single model. All models except "BERT Explanation" are initialized with pre-trained BART weights.

Official EXPECT-*dev* Cor. (P / R / $F_{0.5}$)	Exp. (P / R / F_1 / $F_{0.5}$ / Acc)
30.94 / 35.49 / 31.75	45.92 / 38.42 / 41.84 / 44.19 / 37.63
Rebuilt EXPECT-*dev* **Cor (P / R / $F_{0.5}$)**	**Exp (P / R / F_1 / $F_{0.5}$ / Acc)**
36.34 / 40.15 / **37.05**	48.95 / 42.72 / **45.65** / **47.56** / **40.32**

Table 4: Comparison of *post-explaining* models trained on the official and rebuilt EXPECT datasets. We have similar findings on other settings, which are listed in Appendix B.1.

models on the official and our rebuilt EXPECT datasets. The results in Table 4 indicate that our rebuilt EXPECT dataset can significantly improve the performance of both correction and explanation tasks. This is because we have identified and corrected grammatical errors that were previously overlooked. As a results, we conduct the remaining experiments on the rebuilt EXPECT dataset.

4.3 Main Results

Here, we examine and analyze the interaction between the correction and explanation tasks by conducting experiments with various training settings. We first explore the Infusion setting, where we append different additional explanation information to the input source. Infusion models can be considered as oracle baselines since human-annotated explanations are usually unavailable in real applications, through which we can understand how explanations benefit the correction task. We also train a sequence labeling-based BERT model by reproducing the baseline provided in (Fei et al., 2023) under the same training and evaluation conditions as our other experiments. The results presented in Table 3 illustrate the following conclusions.

Evidence words, rather than grammatical error types, can provide invaluable information for corrections. Recent studies have highlighted that incorporating human-annotated explanations as additional input can enhance task performance to a certain degree (Hase et al., 2020; Yao et al., 2023), and we have also observed similar results in the "Infusion" block of Table 3. Specifically, we notice that the additional information provided by grammatical error types does not improve correction performance. However, on the other hand, the information provided by evidence words can increase the $F_{0.5}$ score by approximately 10 points, even though about only 60% of the samples in the dev and test sets are annotated with evidence words, demonstrating that ground truth evidence words are very helpful for the correction task.

Jointly learning correction and explanation tasks is beneficial for each task. Practically, explanations are usually unavailable during the inference stage, so Self-rationalization models are responsible for answering whether training with explanations as additional output could improve correction performance. Interestingly, experiments show that pre-explaining and post-explaining models perform differently. Specifically, pre-explaining models achieve better correction performance at the cost of decreased explanation performance compared to the "BART Explanation" single-task baseline, demonstrating that even noisy predicted explanations can still provide benefits towards the correction task. On the other hand, post-explaining models achieve comparable correction performance but very high explanation performance, indicating that predicted corrections are very beneficial towards the explanation task.

We also notice that the performance of grammatical error type classification for BART-based models is greatly lower than that of BERT-based models. We speculate that this may be due to the inner bias induced by the distinction between BART's generative denoising and BERT's masked language model

γ	**Cor. (P / R / $F_{0.5}$)**	**Exp. (P / R / F_1 / $F_{0.5}$ / Acc)**
0.5	36.16 / 35.68 / 36.06	57.00 / 06.87 / 12.26 / 23.18 / 19.15
0.8	35.47 / 36.92 / 35.74	51.77 / 21.63 / 30.51 / 40.49 / 23.46
1.0	35.10 / 36.96 / 35.46	48.82 / 26.55 / **34.40** / **41.81** / 25.94
1.5	36.12 / 36.34 / **36.16**	50.95 / 22.01 / 30.74 / 40.34 / 24.66
2.0	35.93 / 35.38 / 35.82	52.48 / 22.29 / 31.29 / 41.29 / **28.06**

Table 5: Results of sequence labeling-based multi-task BART baselines for varying loss weights γ on rebuilt **EXPECT-*dev***.

(MLM) pre-training objectives. This is supported by the experiments in Section 5.1, which indicate that sequence labeling is not the crucial factor for grammatical error type classification.

5 Analysis

5.1 Does Sequence Labeling Help?

Motivated by recent studies in multi-task GEC frameworks (Zhao et al., 2019; Yuan et al., 2021), which combine a sequence labeling task with a sentence-level correction task, we also develop a multi-task baseline for explainable GEC, keeping the experimental setup the same as our other experiments. Specifically, we append a random-initialized tagging head after the encoder to perform the explanation task as a sequence labeling task, like BERT-based models. To predict each token's tag, we pass the encoder's hidden representation $\mathbf{H}^e$ through a softmax after an affine transformation, which is computed as follow:

$$P(T \mid X) = \mathrm{softmax}(W^\top \mathbf{H}^e), \quad (9)$$

where T is the resulting tagging sequence in BIO scheme. The token-level sequence labeling task is introduced to replace the role of pointer mechanism, so we conduct only the correction task at the decoder side. Similarly, we introduce loss weighting to trade-off the losses of correction generation and sequence labeling, which is defined as follow:

$$\mathcal{L} = \mathcal{L}_{cor} + \gamma \cdot \mathcal{L}_{tag} \quad (10)$$

where γ represents the trade-off factor, and we minimize the cross-entropy between predicted tokens/labels and ground truth tokens/labels.

The results of varying γ selected from the alternative set $\{0.5, 0.8, 1.0, 1.5, 2.0\}$ are shown in Table 5. Compared to Self-rationalization models, sequence labeling-based multi-task models achieve lower correction performance but mediate explanation performance between pre-explaining models and post-explaining models. Therefore, we can conclude that our proposed EXGEC is more effective than sequence labeling-base baselines.

5.2 Position Leakage

One may suspect that the enhancement of Infusion models is due to the leakage effect of evidence words' positions, since it is reported that a significant number of instances have at least one evidence word within the first or second-order nodes of correction words in the dependency parsing tree (Fei et al., 2023). To address this concern, we synthesize datasets with artifact explanations in two ways: 1) *random explanations*, which are randomly selected from the entire source tokens, and 2) *adjacent explanations*, which are randomly chosen from candidate source tokens located within a distance of 1$\sim$5 from the correction. Given that a substantial number of samples lack annotated evidence words, we generate an equal number of synthesized evidence words as the ground truth ones to ensure the fairness of our experiments. We train models using synthesized evidence words, but evaluation is performed with ground truth evidence words, allowing us to investigate whether the models learn to extract evidence words through this unsupervised approach.The results are presented in Table 6.

For the Infusion setting, it is no surprise that random evidence words would not improve correction performance as expected. However, we observe that adjacent synthesized evidence words do make a noticeable impact, resulting in a moderate improvement compared to random evidence words but still lower than the benefits provided by ground truth evidence words. This suggests that the leakage effect of positions does indeed exists. However, it is important to note that this effect alone is unable to fully capture all the advantages offered by ground truth evidence words.

For the pre-explaining and post-explaining settings, it seems that learning to output adjacent evidence words can improve correction performance to some extent. However, it falls short of surpassing the performance achieved by incorporating ground truth evidence words. This reaffirms the importance of joint learning for both correction and explanation tasks. On the contrary, the inclusion of random evidence words does not contribute to the improvement of correction performance. Furthermore, the models' explanation performance reveals their inclination to disregard the influence of these random evidence words. Additionally, we observe a significant decrease in explanation per-

System	EXPECT-*dev* Cor. (P / R / $F_{0.5}$)	EXPECT-*dev* Exp. (P / R / F_1 / $F_{0.5}$ / Acc)	EXPECT-*test* Cor. (P / R / $F_{0.5}$)	EXPECT-*test* Exp. (P / R / F_1 / $F_{0.5}$ / Acc)
BART Baseline	36.14 / 34.87 / 35.88	-	36.33 / 35.49 / 36.16	-
Infusion				
+ G.T. Evidence	45.78 / 44.55 / **45.53**	-	46.02 / 44.13 / **45.63**	-
+ Ran. Evidence	35.88 / 33.26 / 35.33	-	36.44 / 33.20 / 35.74	-
+ Adj. Evidence	38.46 / 42.81 / 39.26	-	39.66 / 43.01 / 40.28	-
Pre-explaining				
+ G.T. Evidence	38.25 / 34.18 / **37.36**	36.01 / 35.58 / **35.79** / **35.92** / **26.56**	38.68 / 35.41 / **37.98**	36.77 / 36.85 / **36.81** / **36.79** / **26.24**
+ Ran. Evidence	36.17 / 33.72 / 35.65	13.60 / 00.40 / 00.77 / 01.79 / 15.83	37.63 / 34.83 / 37.04	14.38 / 00.53 / 01.02 / 02.31 / 15.02
+ Adj. Evidence	36.53 / 38.73 / 36.95	26.97 / 03.37 / 06.00 / 11.23 / 17.03	37.09 / 39.52 / 37.55	29.00 / 04.02 / 07.06 / 12.93 / 16.02
Post-explaining				
+ G.T. Evidence	36.34 / 40.15 / **37.05**	48.95 / 42.72 / **45.63** / **47.56** / **40.32**	36.52 / 40.41 / **37.24**	49.43 / 44.10 / **46.61** / **48.26** / **39.86**
+ Ran. Evidence	36.36 / 34.37 / 35.95	14.39 / 00.45 / 00.86 / 02.00 / 16.04	36.86 / 34.87 / 36.44	07.45 / 00.16 / 00.32 / 00.74 / 15.02
+ Adj. Evidence	36.36 / 34.14 / 35.89	23.68 / 02.53 / 04.57 / 08.86 / 15.79	37.34 / 35.18 / 36.88	26.74 / 03.28 / 05.84 / 11.00 / 15.48

Table 6: Results of models trained on ground truth (G.T.), random (Ran.) or adjacent (Adj.) evidence words.

formance when learning without ground truth evidence words, indicating the inherent challenge of explaining with alignment to human preference in an unsupervised way.

6 Related Works

Explainable GEC. Currently, most GEC systems are trained to correct errors without providing explanations. To bridge the gap, recent studies have explored several methods to facilitate the explainability of GEC systems. One such method is the feedback comment generation (FCG) task (Nagata, 2019; Nagata et al., 2021), which is designed to automatically generate feedback comments such as hints or explanatory notes for writing learning. Hanawa et al. (2021) investigate three different architectures for FCG and highlight the challenges of the task. Another approach is Example-based GEC (Kaneko et al., 2022; Vasselli and Watanabe, 2023), which improves explainability by retrieving examples similar to an input instance according to pre-defined grammar rules. Kaneko and Okazaki (2023) explore generating natural language explanations by prompting large language models (LLMs), showing the feasibility of eliciting controlled and comprehensive explanations for grammatical errors from LLMs. However, there has been no work systematically exploring the interaction between correction and explanation tasks.

Learning with Explanations. As an important part of this work, Self-rationalization models jointly generate task predictions and corresponding explanations, aiming to improve explainability or task performance of neural networks. Two approaches that currently predominate the building of self-rationalization models are 1) extracting highlight input tokens responsible for task predictions, known as extractive rationals (DeYoung et al., 2020), and 2) generating natural language explanations (Narang et al., 2020), which provide a natural interface between machine computation and human end-users. To improve upon the task performance and trustworthiness of Seq2Seq models, Lakhotia et al. (2021) develop an extractive fusion-in-decoder architecture in the ERASER benchmark (DeYoung et al., 2020), which is a popular benchmark for rationale extraction across multiple datasets and tasks. Li et al. (2022a) propose a joint text classification and rationale extraction model to improve explainability and robustness. Recognizing the complementarity of extractive rationals and natural language explanations, Majumder et al. (2022) combine both ingredients in a unified self-rationalization framework.

Powered by in-context learning (Brown et al., 2020) and chain-of-thought (CoT) reasoning (Wei et al., 2022; Chu et al., 2023) of LLMs, recent works leverage the natural language explanations generated by LLMs with chain-of-thought prompting (Lampinen et al., 2022; Li et al., 2023) to enhance the training of small reasoners using knowledge distillation for task performance (Li et al., 2022b; Ho et al., 2023; Hsieh et al., 2023) or faithfulness (Wang et al., 2023) improvement.

7 Conclusion

In this paper, we propose a unified generative framework, EXGEC, designed to jointly perform both correction and explanation tasks. EXGEC is designed to be compatible with multiple training settings, enabling us to understand and establish the interaction between tasks. Additionally, we rebuild the existing noisy explainable GEC dataset, EXPECT. Our experiments demonstrate the effectiveness of our rebuild process and EXGEC.

Limitations

Inherent nature of Seq2Seq-based models. We have noticed that our adopted backbone, BART, falls short in explanation performance, including extracting evidence words and classifying grammatical errors, compared to BERT-based models. This can be attributed to BART's inherent nature as a sequence-to-sequence generative model. These limitations may have a negative impact on correction performance, particularly for post-explaining models that correct sentences based on previously predicted explanations. In our future work, we intend to explore a more effective approach to handle and integrate both tasks.

Inflexibility of structured explanations. In the era of large language models (LLMs), it has become increasingly practical and favorable to express explanations as free-form natural language texts. However, in this particular paper, we focus our studies on structured explanations due to the limited availability of free-form explanations in the field of GEC. Nevertheless, we are committed to advancing the development of explainable GEC datasets in our future work, aiming to incorporate more sophisticated and comprehensive approaches.

Ethics Statement

In this paper, we have identified significant noise in the official EXPECT dataset, which has the potential to create confusion during model training and evaluation. To address this issue, we reconstruct the EXPECT dataset to remove the noise, resulting in an objective training and evaluation pipeline. For our methods, we have exclusively utilized source data from publicly accessible project resources on legitimate websites, ensuring the absence of sensitive information. Furthermore, all the baselines and datasets utilized in our experiments are publicly available, and we have given credit to the corresponding authors by citing their work.

References

John Bitchener, Stuart Young, and Denise Cameron. 2005. The effect of different types of corrective feedback on esl student writing. *Journal of second language writing*, 14(3):191–205.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2022. Grammatical error correction: A survey of the state of the art. *arXiv preprint arXiv:2211.05166*.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. 2023. A survey of chain of thought reasoning: Advances, frontiers and future. *arXiv preprint arXiv:2309.15402*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Tao Fang, Jinpeng Hu, Derek F. Wong, Xiang Wan, Lidia S. Chao, and Tsung-Hui Chang. 2023. Improving grammatical error correction with multimodal feature integration. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9328–9344, Toronto, Canada. Association for Computational Linguistics.

Yuejiao Fei, Leyang Cui, Sen Yang, Wai Lam, Zhenzhong Lan, and Shuming Shi. 2023. Enhancing grammatical error correction systems with explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7489–7501, Toronto, Canada. Association for Computational Linguistics.

Shichun Gui. 2004. A cognitive model of corpus-based analysis of chinese learners' errors of english. *Modern Foreign Languages(Quarterly)*, 27(2):129–139.

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.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367, Online. Association for Computational Linguistics.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada. Association for Computational Linguistics.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 595–606, New Orleans, Louisiana. Association for Computational Linguistics.

Masahiro Kaneko and Naoaki Okazaki. 2023. Controlled generation with prompt insertion for natural language explanations in grammatical error correction. *arXiv preprint arXiv:2309.11439*.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kushal Lakhotia, Bhargavi Paranjape, Asish Ghoshal, Scott Yih, Yashar Mehdad, and Srini Iyer. 2021. FiD-ex: Improving sequence-to-sequence models for extractive rationale generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3712–3727, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. 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.

Dongfang Li, Baotian Hu, Qingcai Chen, Tujie Xu, Jingcong Tao, and Yunan Zhang. 2022a. Unifying model explainability and robustness for joint text classification and rationale extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10947–10955.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also" think" step-by-step. *arXiv preprint arXiv:2306.14050*.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, et al. 2022b. Explanations from large language models make small reasoners better. *arXiv preprint arXiv:2210.06726*.

Josh Magnus Ludan, Yixuan Meng, Tai Nguyen, Saurabh Shah, Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2023. Explanation-based finetuning makes models more robust to spurious cues. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4420–4441, Toronto, Canada. Association for Computational Linguistics.

Bodhisattwa Prasad Majumder, Oana Camburu, Thomas Lukasiewicz, and Julian Mcauley. 2022. Knowledge-grounded self-rationalization via extractive and natural language explanations. In *International Conference on Machine Learning*, pages 14786–14801. PMLR.

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.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *arXiv preprint arXiv:2004.14546*.

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. *arXiv preprint arXiv:1904.01038*.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Younghee Sheen. 2007. The effect of focused written corrective feedback and language aptitude on esl learners' acquisition of articles. *TESOL quarterly*, 41(2):255–283.

Apoorva Singh, Raghav Jain, Prince Jha, and Sriparna Saha. 2023. Peeking inside the black box: A commonsense-aware generative framework for explainable complaint detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7333–7347, Toronto, Canada. Association for Computational Linguistics.

Peter Skehan. 1998. *A cognitive approach to language learning*. Oxford University Press.

Justin Vasselli and Taro Watanabe. 2023. A closer look at k-nearest neighbors grammatical error correction. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 220–231, Toronto, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Aditya Srikanth Veerubhotla, Lahari Poddar, Jun Yin, György Szarvas, and Sharanya Eswaran. 2023. Few shot rationale generation using self-training with dual teachers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4825–4838, Toronto, Canada. Association for Computational Linguistics.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems*, 28.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. 2023. SCOTT: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5546–5558, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5808–5822, Online. Association for Computational Linguistics.

Bingsheng Yao, Prithviraj Sen, Lucian Popa, James Hendler, and Dakuo Wang. 2023. Are human explanations always helpful? towards objective evaluation of human natural language explanations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14698–14713, Toronto, Canada. Association for Computational Linguistics.

Zheng Yuan and Christopher Bryant. 2021. Document-level grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 75–84, Online. Association for Computational Linguistics.

Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 2021. Multi-class grammatical error detection for correction: A tale of two systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8722–8736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yue Zhang, Bo Zhang, Haochen Jiang, Zhenghua Li, Chen Li, Fei Huang, and Min Zhang. 2023. NaSGEC: a multi-domain Chinese grammatical error correction dataset from native speaker texts. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9935–9951, Toronto, Canada. Association for Computational Linguistics.

Yue Zhang, Bo Zhang, Zhenghua Li, Zuyi Bao, Chen Li, and Min Zhang. 2022. SynGEC: Syntax-enhanced grammatical error correction with a tailored GEC-oriented parser. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2518–2531, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2023. Explainability for large language models: A survey. *arXiv preprint arXiv:2309.01029*.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. 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 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

A Experiment Hyper-Parameters

We list the main hyper-parameters in Table 7. For the training stage, we follow the same hyper-parameters as described in (Zhang et al., 2022). The total training time is about 4 hours.

Configuration	Value
Training	
Backbone	BART-large (Lewis et al., 2020)
Devices	1 Tesla A100 GPU (80GB)
Epochs	60
Batch size per GPU	4096 tokens
Gradient Accumulation	4
Optimizer	Adam (Kingma and Ba, 2014) ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8}$)
Learning rate	3×10^{-5}
Warmup updates	500
Max source length	256
Dropout	0.3
Dropout-src	0.1
α	0.5
Loss weight λ	1.0
Inference	
Beam size	12
Max length	256

Table 7: Hyper-parameter values used in our experiments.

B Extra Analyses

B.1 Detailed Results on EXPECT Datasets

We report the detailed results on the official and our rebuilt EXPECT dasetes in Table 9. All the models trained on our rebuilt EXPECT achieve better performance of both correction and explanation tasks, demonstrating the effectiveness of our rebuild process.

B.2 Impact of Loss Weighting

In this section, we investigate the trade-off of learning on both correction and explanation task by varying the loss weight λ. Considering the promising performance of post-explaining models on both correction and explanation tasks, we train post-explaining models with the loss weight λ alternatively selected from $\{0.5, 1.0, 1.5, 2.0\}$ and report the results on EXPECT-*dev* in Table 8. The results show that giving preference to either tasks harms the performance of both tasks. We speculate that the supervised explanation information during training is too weak to guide the dynamics of correction learning if λ is small. On the other hand, a large λ value might neglect correction learning, thus leading to lower explanation performance since explanation of post-explaining models are produced based on predicted corrections.

λ	**Cor. (P / R / F$_{0.5}$)**	**Exp. (P / R / F$_1$ / F$_{0.5}$ / Acc)**
0.5	35.40 / 38.03 / 35.90	39.77 / 38.88 / 39.32 / 39.59 / 32.02
1.0	36.34 / 40.15 / **37.05**	48.95 / 42.72 / **45.63** / **47.56** / **40.32**
1.5	36.03 / 38.42 / 36.49	43.90 / 42.82 / 43.35 / 43.68 / 36.88
2.0	35.41 / 38.61 / 36.00	47.98 / 42.86 / 45.28 / 46.86 / 40.07

Table 8: Results of *post-explaining* models for varying loss weights λ on rebuilt **EXPECT-*dev***.

	Official EXPECT-*dev*		Rebuilt EXPECT-*dev*	
System	**Cor. (P / R / F$_{0.5}$)**	**Exp. (P / R / F$_1$ / F$_{0.5}$ / Acc)**	**Cor. (P / R / F$_{0.5}$)**	**Exp. (P / R / F$_1$ / F$_{0.5}$ / Acc)**
BART Baseline	30.59 / 33.72 / 31.17	-	36.14 / 34.87 / **35.88**	-
Infusion				
+ Evidence	40.72 / 43.31 / 41.22	-	45.78 / 44.55 / **45.53**	-
+ Type	31.15 / 35.14 / 31.87	-	35.31 / 47.87 / 35.22	-
+ Evidence&Type	40.79 / 42.50 / 41.11	-	44.28 / 47.55 / 44.90	-
Self-rationalization				
Pre-explaining	32.62 / 31.29 / 32.35	33.75 / 44.12 / 38.25 / 35.41 / 28.22	38.25 / 34.18 / **37.36**	36.01 / 35.58 / 35.79 / 35.92 / 26.56
Post-explaining	30.94 / 35.49 / 31.75	45.92 / 38.42 / 41.84 / 44.19 / 37.63	36.34 / 40.15 / 37.05	48.95 / 42.72 / **45.63** / **47.56** / **40.32**

Table 9: Further comparison of models trained on the official and rebuilt EXPECT datasets.