Enhancing Grammatical Error Correction Systems with Explanations

Yuejiao Fei[♠♣*], **Leyang Cui**[♡†], **Sen Yang**[♢], **Wai Lam**[♢], **Zhenzhong Lan**[♣], **Shuming Shi**[♡]
[♠] Zhejiang University
[♡] Tencent AI lab
[♢] The Chinese University of Hong Kong
[♣] School of Engineering, Westlake University
{feiyuejiao,lanzhenzhong}@westlake.edu.cn
{leyangcui,shumingshi}@tencent.com
{syang, wlam}@se.cuhk.edu.hk

Abstract

Grammatical error correction systems improve written communication by detecting and correcting language mistakes. To help language learners better understand why the GEC system makes a certain correction, the causes of errors (evidence words) and the corresponding error types are two key factors. To enhance GEC systems with explanations, we introduce EXPECT, a large dataset annotated with evidence words and grammatical error types. We propose several baselines and analysis to understand this task. Furthermore, human evaluation verifies our explainable GEC system's explanations can assist second-language learners in determining whether to accept a correction suggestion and in understanding the associated grammar rule.

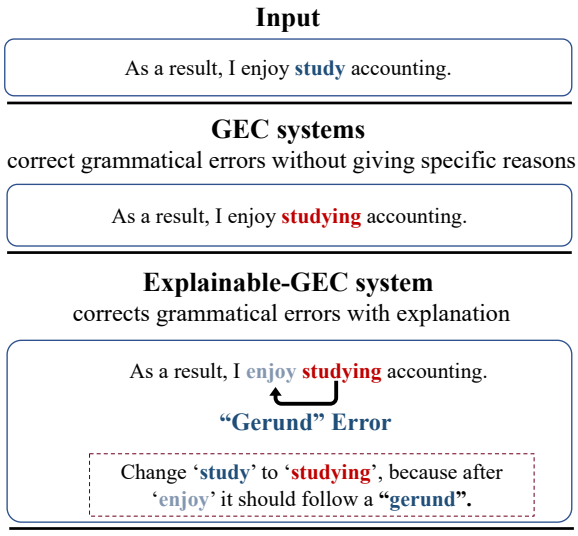

Figure 1: Comparison between explainable GEC and conventional GEC systems.

1 Introduction

Grammatical Error Correction (GEC) systems aim to detect and correct grammatical errors in a given sentence and thus provide useful information for second-language learners. There are two lines of work for building GEC systems. Sequence-to-sequence methods (Rothe et al., 2021; Flachs et al., 2021; Zhang et al., 2022) take an erroneous sentence as input and generate an error-free sentence autoregressively. Sequence labeling methods (Omelianchuk et al., 2020; Tarnavskyi et al., 2022a) transform the target into a sequence of text-editing actions and use the sequence labeling scheme to predict those actions.

With advances in large pre-trained models (Devlin et al., 2018; Lewis et al., 2019) and availability of high-quality GEC corpora (Ng et al., 2014; Bryant et al., 2019), academic GEC systems (Omelianchuk et al., 2020; Rothe et al., 2021) have achieved promising results on benchmarks and serve as strong backbones for modern writing assistance applications (e.g., Google Docs[1], Grammarly[2], and Effidit (Shi et al., 2023)[3]). Although these academic methods provide high-quality writing suggestions, they rarely offer explanations with specific clues for corrections. Providing a grammar-aware explanation and evidence words to support the correction is important in second-language education scenarios (Ellis et al., 2006), where language learners need to "know why" than merely "know how". As a commercial system, Grammarly does provide evidence words, but in very limited cases, and the technical details are still a black box for the research community.

Though some existing work has attempted to enhance the explainability of GEC's corrections (Bryant et al., 2017; Omelianchuk et al., 2020; Kaneko et al., 2022), they do not provide intra-sentence hints (i.e., evidence words in the sentence). To fill this gap, we build a dataset

*Work was done during the internship at Tencent AI lab.
†Corresponding authors.

[1] https://www.google.com/docs/about/
[2] https://demo.grammarly.com/
[3] https://effidit.qq.com/

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7489–7501
July 9-14, 2023

named **EXP**lainble grammatical **E**rror **C**orrec**T**ion (**EXPECT**) on the standard GEC benchmark (W&I+LOCNESS (Bryant et al., 2019)), yielding 21,017 instances with explanations in total. As shown in Figure 1, given a sentence pair consisting of an erroneous sentence and its corrected counterpart, our explainable annotation includes:

1) **Evidence words** in the erroneous sentence. Error tracing could be rather obscure for second-language beginners. For example, given an erroneous sentence, "*As a result, I enjoy study accounting.*" where "*study*" should be corrected to "*studying*", a beginning learner might mistakenly attribute "*studying*" to "*accounting*" because they both have an "*ing*" suffix. However, the correct attribution should be "*enjoy*". Such incorrect judgment may lead the language learner to draw wrong conclusions (e.g., A verb needs to have an "*ing*" suffix if a subsequent verb does so), which significantly disturbs the learning procedure. To remedy this, EXPECT provides annotated evidence words, which enable training models to automatically assist second-language learners in finding error clues.

2) **Error types** of the grammatical errors, ranging among the 15 well-defined categories by consulting the pragmatic errors designed by Skehan et al. (1998) and Gui (2004). Language learning consists of both abstract grammar rules and specific language-use examples. A model trained with EXPECT bridges the gap between the two parts: such a model can produce proper error types, automatically facilitating language learners to infer abstract grammar rules from specific errors in an inductive reasoning manner. Further, it allows learners to compare specific errors within the same category and those of different categories, benefiting the learner's inductive and deductive linguistic reasoning abilities.

To establish baseline performances for explainable GEC on EXPECT, we explore generation-based, labeling-based, and interaction-based methods. Note that syntactic knowledge also plays a crucial role in the human correction of grammatical errors. For example, the evidence word of subject-verb agreement errors can be more accurately identified with the help of dependency parsing. Motivated by these observations, we further inject the syntactic knowledge produced by an external dependency parser into the explainable GEC model.

Experiments show that the interaction-based method with prior syntactic knowledge achieves the best performance ($F_{0.5}$ =70.77). We conduct detailed analysis to provide insights into developing and evaluating an explainable GEC system. Human evaluations suggest that the explainable GEC systems trained on EXPECT can help second language learners to understand the corrections better. We will release EXPECT (e.g., baseline code, model, and human annotations) on `https://github.com/lorafei/Explainable_GEC`.

2 Related Work

Some work formulates GEC as a sequence-to-sequence problem. Among them, transformer-based GEC models (Rothe et al., 2021) have attained state-of-the-art performance on several benchmark datasets (Ng et al., 2014; Bryant et al., 2019) using large PLMs (Raffel et al., 2020) and synthetic data (Stahlberg and Kumar, 2021). To avoid the low-efficiency problem of seq2seq decoding, some work (Awasthi et al., 2019; Omelianchuk et al., 2020; Tarnavskyi et al., 2022b) formats GEC as a sequence labeling problem and achieves competitive performance. Both lines of work focus on improving the correction performance and decoding speed but cannot provide users with further suggestions.

Several methods have been proposed to provide explanations for GEC systems. ERRANT (Bryant et al., 2017) designs a rule-based framework as an external function to classify the error type information given a correction. GECToR (Omelianchuk et al., 2020) pre-defines g-transformations tag (e.g., transform singular nouns to plurals) and uses a sequence labeling model to predict the tag as explanations directly. Example-based GEC (Kaneko et al., 2022) adopts the k-Nearest-Neighbor method (Khandelwal et al., 2019) for GEC, which can retrieve examples to improve interpretability. Despite their success, their explanations are restricted by pre-defined grammar rules or unsupervised retrieval. They may not generalize well to real-life scenarios due to the limited coverage of the widely varying errors made by writers. In contrast, our annotated instances are randomly sampled from real-life human-written corpora without restriction, thus providing a much larger coverage.

Nagata (2019); Nagata et al. (2020); Hanawa et al. (2021), and Nagata et al. (2021) propose a feedback comment generation task and release two corresponding datasets, which, to our knowledge, are the only two publicly available and large-scale datasets focusing on GEC explanations. The task aims to generate a fluent comment describing the erroneous sentence's grammatical error. While this task integrates GEC and Explainable-GEC into one text generation task, we only focus on Explainable-GEC and formulate it as a labeling task, which is easier and can avoid the high computational cost of seq2seq decoding. Furthermore, the evaluation of feedback comment generation mainly relies on human annotators to check if the error types are correctly identified and if the grammatical error correction is proper in the generated text, which is time-consuming and susceptible to the variations resulting from subjective human judgment. In contrast, our token classification task can be easily and fairly evaluated by automatic metrics (e.g., F-score), favoring future research in this direction.

3 Dataset

To facilitate more explainable and instructive grammatical error correction, we propose the EXPECT, an English grammatical error explanation dataset annotated with 15 grammatical error types and corresponding evidence words.

3.1 Data Source

We annotate EXPECT based on W&I+LOCNESS (Bryant et al., 2019), which comprises 3,700 essays written by international language learners and native-speaking undergraduates and corrected by English teachers. We first select all sentences with errors from essays. For a sentence with n errors, we repeat the sentence n times and only keep a single unique error in each sentence. Then, we randomly sample and annotate 15,187 instances as our training set. We do the same thing for the entire W&I+LOCNESS dev set, and split it up into test and development sets evenly.

In order to better align with real-world application scenarios, we have additionally annotated 1,001 samples based on the output of the conventional GEC models. We randomly sampled the output of T5-large (Rothe et al., 2021) and GECToR-Roberta (Omelianchuk et al., 2020) on the W&I+LOCNESS test set. We also report whether the corrections of the GEC model were right.

3.2 Error Type Definition

Following the cognitive model of second language acquisition (Skehan et al., 1998; Gui, 2004), we design error types among three cognitive levels as follows:

Single-word level error is in the first and lowest cognitive level. These mistakes usually include misuse of *spelling*, *contraction*, and *orthography*, which are often due to misremembering. Since there is no clear evidence for those errors, we classify them into type *others*.

Inter-word level error is in the second cognitive level, which usually stems from a wrong understanding of the target language. Most error types with clear evidence lie at this level because it represents the interaction between words. This level can be further split into two linguistic categories, syntax class and morphology class: (1) In the view of syntax, we have seven error types, including *infinitives*, *gerund*, *participles*, *subject-verb agreement*, *auxiliary verb*, *pronoun* and *noun possessive*. (2) In the view of morphology, we have five error types, including *collocation*, *preposition*, *word class confusion*, *numbers*, and *transitive verbs*.

Discourse level error is at the highest cognitive level, which needs a full understanding of the context. These errors include *punctuation*, *determiner*, *verb tense*, *word order* and *sentence structure*. Since *punctuation*, *word order*, and *sentence structure* errors have no clear evidence words, we also classify them into type *others*.

The complete list of error types and corresponding evidence words are listed in Figure 2. The definition of each category is shown in Appendix A.1.

3.3 Annotation Procedure

Our annotators are L2-speakers who hold degrees in English and linguistics, demonstrating their proficiency and expertise in English. The data are grouped into batches of 100 samples, each containing an erroneous sentence and its correction. The annotators are first trained on labeled batches until their F_1 scores are comparable to those of the main author. After that, annotators are asked to classify the type of the correction and highlight *evidence words* that support this correction on the unlabeled batches. The evidence should be **informative** enough to support the underlying grammar of the correction meanwhile **complete** enough to

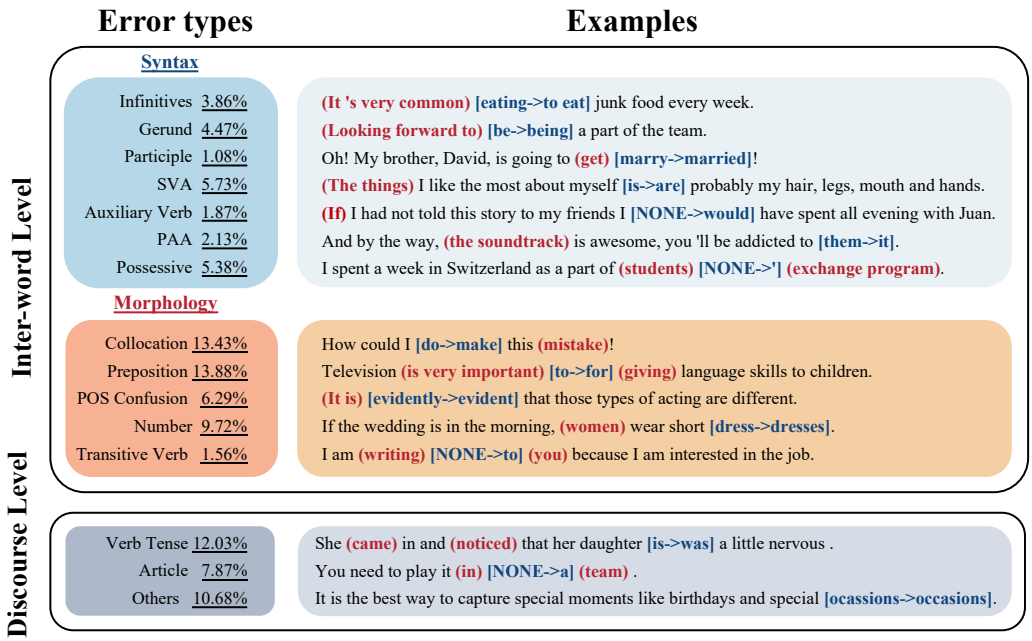

Figure 2: Examples of each error type and corresponding evidence words in EXPECT. Blue text indicates the correction, while red text indicates the evidence words. SVA means subject-verb agreement, PAA means pronoun-antecedent agreement, POS confusion means part-of-speech confusion.

Data Statistics	Train	Dev	Test	Outputs
Number of sentences	15,187	2,413	2,416	1001
Number of words	435,503	70,111	70,619	27,262
Avg. w.p.s	28.68	29.06	29.23	27.23
With evidence rate	74.15	59.10	59.77	72.73
Total evidence words	29,187	4,280	4,340	1736
Avg. evidence w.p.s	2.59	3.00	3.01	2.38

Table 1: Data Statistics of EXPECT. Here w.p.s means word per sentence.

include all possible evidence words. For each complete batch, we have an experienced inspector to re-check 10% of the batch to ensure the annotation quality. According to inspector results, if F_1 scores for the annotation are lower than 90%, the batch is rejected and assigned to another annotator.

3.4 Data Statistics

The detailed statistics of EXPECT have listed in Table 1. Take the train set for example, the average number of words per sentence is 28.68, and 74.15% of the entire dataset has explainable evidence. Among all sentences with evidence words, the average number of words per evidence is 2.59. The percentage of all error types is listed in Figure 2. Detailed description for all error categories is listed in Appendix A.1. Top-3 most frequent error types are *preposition* (13.88%), *collocation* (13.43%) and *verb tense* (12.03%).

Precision	Recall	F_1	$F_{0.5}$	Exact Match
0.469	0.410	0.463	**0.471**	0.342

Table 2: Pearson correlation between human judgment and different automatic evaluation metrics.

3.5 Evaluation Metrics

We consider our task as a token classification task. Thus we employ token-level (precision, recall, F_1, and $F_{0.5}$) and sentence-level (exact match, label accuracy) evaluation metrics. Specifically, the **exact match** requires identical error types and evidence words between label and prediction, and the **label accuracy** measures the classification performance of error types. To explore which automatic metric is more in line with human evaluation, we compute Pearson correlation (Freedman et al., 2007) between automatic metrics and human judgment. As shown in Table 2, $F_{0.5}$ achieves the highest score in correlation. And precision is more correlated with human judgment than recall. The reason may be that finding the precise evidence words is more instructive than extracting all evidence words for explainable GEC.

4 Methods

In this section, we define the task of explainable-GEC in Section 4.1 and then introduce the labeling-

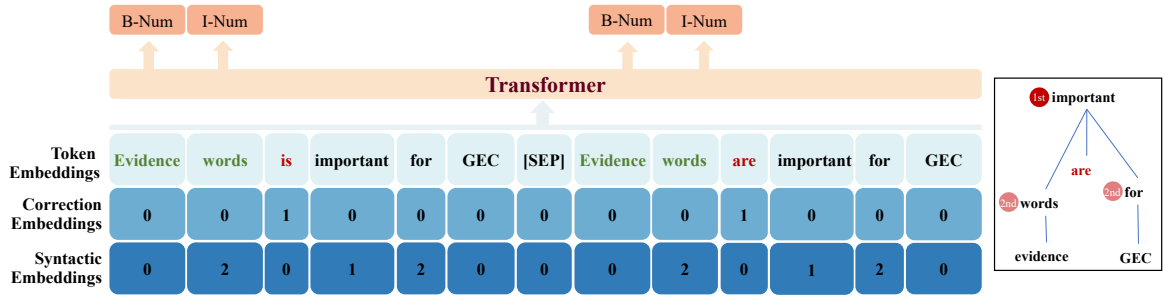

Figure 3: An illustration of labeling-based methods with syntax for solving explainable GEC. On the right is the dependency parsing tree of the corrected sentence, where the correction word *are* is marked in red, and 1st and 2nd-order nodes are marked with red circles.

based baseline method in Section 4.2, and the interaction method in Section 4.3.

4.1 Task Formulation

The task input is a pair of sentences, including an erroneous sentence $X = \{x_1, x_2, ..., x_n\}$ and the corresponding corrected sentence $Y = \{y_1, y_2, ..., y_m\}$. The two sentences usually share a large ratio of overlap. The difference between the two sentences is defined as a span edit $\{(s_x, s_y)\}$. The task of explainable GEC is to find the grammar evidence span E_x within X and predict corresponding error type classes c. Take Figure 3 as an example, $s_x =$ "are" and $s_y =$ "is", the evidence span $E_x =$"Evidence words".

4.2 Labeling-based Method

We adopt the labeling-based method for explainable GEC.

Input. We concatenate the erroneous sentence X and the corresponding error-free sentence Y, formed as `[CLS]`X`[SEP]`Y`[SEP]`.

Correction Embedding. To enhance the positional information of the correction, we adopt a correction embedding e_c to encode the position of the correction words in the sentence X and Y. We further add $\mathbf{e}_c$ to embeddings in BERT-based structure as follow:

$$\mathbf{e} = \mathbf{e}_t + \mathbf{e}_p + \mathbf{e}_c \quad (1)$$

where $\mathbf{e}_t$ is the token embeddings, and $\mathbf{e}_p$ is the position embeddings.

Syntactic Embedding. There is a strong relation between evidence words and syntax as shown in Section 5.3. Hence we inject prior syntactic information into the model. Firstly, given the corrected sentence Y and its span edit (s_x, s_y), we parse sentence Y with an off-the-shell dependency parser from the AllenNLP library (Gardner et al., 2018). For each word in s_y, we extract its first-order dependent and second-order dependent words in the dependency parse tree. For example, as shown in Figure 3, the correction word $s_y =$ "are", the first-order dependent word is "important", and the second-order dependent words are "words", and "for", and they are marked separately. By combining all correction edits' first-order words and second-order words, we construct the syntactic vector $d_Y \in \mathbb{R}^m$ for sentence Y. Dependency parsing is originally designed for grammatical sentences. To acquire the syntax vector of the erroneous sentence X, we use the word alignment to map the syntax-order information from the corrected sentence to the erroneous sentence, yielding $d_X \in \mathbb{R}^n$. We then convert $[d_X, d_Y]$ to syntactic embedding $\mathbf{e}_s$, and add to the original word embedding:

$$\mathbf{e} = \mathbf{e}_t + \mathbf{e}_p + \mathbf{e}_c + \mathbf{e}_s \quad (2)$$

Encoder. We adopt a pre-trained language model (e.g. BERT) as an encoder to encode the input $\mathbf{e}$, yielding a sequence of hidden representation $\mathbf{H}$.

Label Classifier. The hidden representation $\mathbf{H}$ is fed into a classifier to predict the label of each word. The classifier is composed of a linear classification layer with a softmax activation function.

$$\hat{l}_i = \text{softmax}(\mathbf{W}\mathbf{h}_i + \mathbf{b}), \quad (3)$$

where $\hat{l}_i$ is the predicted label for i-th word, $\mathbf{W}$ and $\mathbf{b}$ are the parameters for the softmax layer.

Training. The model is optimized by the log-likelihood loss. For each sentence, the training object is to minimize the cross-entropy between l_i

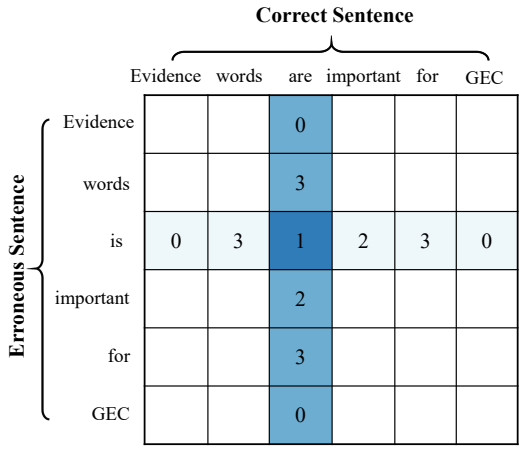

Figure 4: Syntactic Interactive Matrix.

and $\hat{l}_i$ for a labeled gold-standard sentence.

$$\mathcal{L} = -\sum_{i}^{m+n+1} \log \hat{l}_i. \tag{4}$$

4.3 Interaction-based Method

Although labeling-based methods model the paired sentences in a joint encoder, it still predicts two separate outputs independently. The dependencies between the erroneous sentence and the corrected sentence are not explicitly modeled. Intuitively, the alignment between the erroneous sentence and the corrected sentence can be highly informative. We propose an interactive matrix to jointly model the alignment and the evidence span. In particular, we adopt a bi-affine classifier to model the multiplicative interactions between the erroneous sentence and the corrected sentence. Assume that the hidden representation of the erroneous sentence and the corrected sentence are $\mathbf{H}^e$ and $\mathbf{H}^c$, respectively.

We first use two separate feed-forward networks to map the hidden representation into an erroneous query representation and a corrected key representation:

$$\begin{aligned} \mathbf{H}^q &= \mathbf{W}^q \mathbf{H}^e + \mathbf{b}^e \\ \mathbf{H}^k &= \mathbf{W}^k \mathbf{H}^c + \mathbf{b}^c \end{aligned} \tag{5}$$

Then a bi-affine attention (Dozat and Manning, 2016) is adopted to model the interaction between $\mathbf{H}^q$ and $\mathbf{H}^k$:

$$\hat{\mathbf{M}} = softmax(\mathbf{H}^q \mathbf{U} \mathbf{H}^k + \mathbf{b}^U), \tag{6}$$

where $\mathbf{U} \in \mathbb{R}^{|H| \times |H| \times |L|}$, $|H|$ and $|L|$ indicates the hidden size and the size of the label set.

Training. Similar to the labeling-based method, the training objective is to minimize the cross-entropy between $\mathbf{M}$ and $\hat{\mathbf{M}}$ given a labeled gold-standard sentence:

$$\mathcal{L} = -\sum_{i}^{m} \sum_{j}^{n} \log \hat{M}_{ij}. \tag{7}$$

Syntactic Interactive Matrix. Similar to *Syntactic Embedding*, we use a syntactic interactive matrix to better merge the syntactic knowledge into the model. We construct the syntactic interactive matrix $\mathbf{D}^{syn}$ in the same way as the syntactic embedding above, except for using a interactive matrix rather than a flat embedding. Figure 4 shows an example of a syntactic matrix, where the row of the correction index in the erroneous sentence is placed with a syntactic vector of the corrected sentence, whereas the column of the correction index in a corrected sentence is placed with erroneous sentence's syntactic vector. Then a two-layer MLP is used to map $\mathbf{D}^{syn}$ to $\mathbf{H}^{syn}$:

$$\mathbf{H}^{syn} = \mathbf{W}_2^{\text{syn}} \text{ReLU}(\mathbf{W}_1^{\text{syn}} \mathbf{D}^{syn} + \mathbf{b}_1^{\text{syn}}) + \mathbf{b}_2^{\text{syn}} \tag{8}$$

$\mathbf{H}^{syn}$ is then used as an auxiliary term to calculate the interaction matrix $\mathbf{M}$. Eq 6 is reformulated as:

$$\hat{\mathbf{M}} = softmax(\mathbf{H}^q \mathbf{U} \mathbf{H}^k + \mathbf{H}^{syn} + \mathbf{b}^U). \tag{9}$$

5 Experiments

5.1 Baseline Methods

Human performance is reported. We employ three NLP researchers to label the test set and report the average score as human performance.

Generation-based method frames the task as a text generation format. It utilizes a pre-trained generation model to predict the type of error and generate a corrected sentence with highlighted evidence words marked by special tokens.

Labeling-based (***error only***) method uses only erroneous sentences as input and predicted explanation directly.

Labeling-based (***correction only***) method uses only corrected sentences as input and predicted explanation directly.

Labeling-based (***with appendix***) method uses only erroneous sentences or corrected sentences and appends correction words at the end of the sentence.

Labeling-based (***error and correction***) method concatenate erroneous and corrected sentences as described in Section 4.2.

Methods	Dev						Test					
	P	R	F_1	$F_{0.5}$	EM	Acc	P	R	F_1	$F_{0.5}$	EM	Acc
Human	-	-	-	-	-	-	77.50	75.98	76.73	77.19	69.00	87.00
Generation-based												
BART-large	65.75	62.16	63.91	65.00	49.73	75.96	65.68	**61.98**	63.78	64.90	49.20	79.12
Labeling-based												
Error only	50.39	33.41	40.18	45.74	39.77	56.13	50.31	35.07	41.33	46.29	39.68	56.06
Correction only	24.77	14.07	17.94	21.50	29.34	37.34	23.14	12.53	16.26	19.79	28.97	37.67
Error+Appendix	62.92	58.36	60.55	61.95	47.85	72.33	64.78	60.81	62.73	63.94	47.91	73.27
Correction+Appendix	64.85	55.74	59.95	62.80	50.00	74.36	61.86	54.45	57.92	60.22	47.66	72.98
Error+Correction	67.82	57.51	62.24	65.47	50.60	72.42	68.91	57.94	62.95	66.39	59.19	77.31
Error+Correction+CE	69.76	62.20	65.77	68.11	54.09	75.65	69.44	60.93	**64.91**	67.55	61.39	79.14
Error+Correction+CE+Syntax	70.06	**62.44**	**66.03**	68.39	55.21	76.57	68.23	61.23	64.54	66.71	61.26	78.93
Interaction-based												
Error+Correction+CE	71.63	59.54	65.03	68.83	63.04	80.05	68.47	59.14	63.46	66.38	66.28	81.17
Error+Correction+CE+Syntax	**74.77**	58.31	65.52	**70.77**	**64.58**	**81.34**	**73.05**	56.45	63.69	**68.99**	**67.81**	**81.79**

Table 3: Model performance on EXPECT. EM means Exact Match, CE means correction embeddings.

5.2 Main Results

The model performance under different settings are shown in Table 3.

We evaluate the model performance across a variety of settings, including generation-based, labeling-based, and interaction-based, as well as syntactic-based and non-syntactic-based. First, we find that generation-based methods do not outperform labeling-based methods and suffer from poor inference efficiency due to auto-regressive decoding. In addition, interaction-based methods exhibit higher precision but lower recall compared to labeling-based methods. This is likely due to the interaction between two sentences helping the model identify more evidence words. Based on labeling-based methods, adding syntactic information has a marginal 0.28 $F_{0.5}$ point increase, while for interaction-based methods, the performance increases by 1.94 $F_{0.5}$ point. This suggests that syntactic information can generally provide an indication for identifying evidence words. And the interaction matrix better incorporates syntactic information into the model. Particularly, we found correction embeddings are pretty important for this task. With correction embeddings, the performance increases by 2.64 $F_{0.5}$ points on Dev set and 1.16 points on Test set. Finally, interaction-based methods with syntactic knowledge achieve the best performance when measured by precision, $F_{0.5}$, exact match, and accuracy.

5.3 Impact of Syntactic Knowledge

To further explore the role of syntactic knowledge in boosting the explainable GEC performance, we first analyze the relation between evidence words and correction words' adjacent nodes in the dependency parsing tree. As shown in Table 4, 46.71% of instances have at least one evidence word within correction words' first-order nodes, and 27.02% of instances' all evidence words stay within second-order nodes. We can infer that syntactic knowledge can in a way narrow the search space of extracting evidence words.

	Count	Ratio
Exist evidence word in 1st	7,094	46.71
Exist evidence word in 2st	7,723	50.85
All evidence words in 1st	2,528	16.65
All evidence words in 2st	4,103	27.02

Table 4: Statistics of training set evidence words within first-order and second-order nodes.

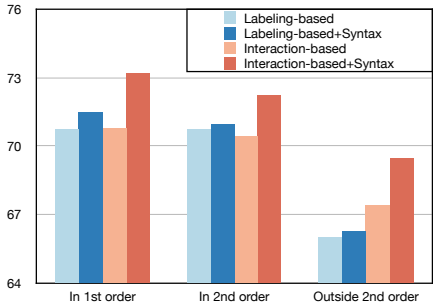

Figure 5: $F_{0.5}$ score comparison of evidence words in first and second order nodes.

Model Performance across Syntactic Distance. We compare $F_{0.5}$ scores for instances whose evidence words are in and out of the 1st and 2nd dependent orders in Figure 5. The overall performance decreases when evidence words are outside the 2nd dependent order, indicating that the model has trouble in handling complex syntactic structure. But after injecting the syntactic knowledge, the performance increases in all sections, suggesting the effectiveness of syntactic representation.

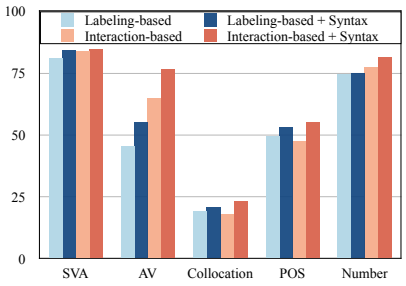

Figure 6: $F_{0.5}$ score comparison of syntax-related error types between syntactic methods and non-syntactic methods. POS - POS Confusion.

Sentence length	#Samples	Labeling -based	Labeling-based + Syntax	Interactive -based	Interaction-based + Syntax
Less than 10	160	72.15	73.52	71.43	**77.57**
10 to 20	751	70.44	69.96	68.22	**71.09**
20 to 30	730	67.57	67.17	70.27	**71.68**
30 to 40	376	66.86	**69.63**	67.25	69.28
40 to 60	239	66.86	66.88	67.01	**68.80**
More than 60	157	62.45	62.25	**70.36**	64.47

Table 5: Model performance $F_{0.5}$ scores across sentence length.

Benefit of Syntactic Representation. We report $F_{0.5}$ scores on specific error types before and after injecting syntactic information into the models in Figure 6. Dependency parsing is a common tool to detect *SVA*(Sun et al., 2007). The performance on *SVA* indeed increases with the syntax. We also find four other error types which are closely associated with syntactic information, including *auxiliary verb*, *collocation*, *POS confusion* and *number*, whose performance increases significantly for both the labeling-based method and the interaction-based method.

5.4 Impact of Sentence Length

Table 5 illustrates the model performance across different lengths of erroneous sentences. As the sentence length increases, the performance of all methods decreases significantly, which is consistent with human intuition. Longer sentences may contain more complex syntactic and semantic structures, which are challenging for models to capture.

5.5 Result on Real-world GEC System

We employ the gold correction as the input during both the training phase and the inference phase. However, in a practical scenario, this input would be replaced with the output of a GEC system. To evaluate the performance of the explainable system equipped with real-world GEC systems, we use interaction-based methods with syntactic knowledge trained on EXPECT, and directly test using samples that are annotated from the outputs of the GEC model on the W&I+LOCNESS test set. The $F_{0.5}$ scores obtained are 57.43 for T5-large outputs and 60.10 for GECToR-Roberta outputs, which significantly underperforms 68.39. This may be caused by the training-inference gap as mentioned and the error propagation of the GEC system.

5.6 Human Evaluation

To assess the effectiveness of the explainable GEC for helping second-language learners understand corrections, we randomly sample 500 instances with gold GEC correction and 501 outputs decoded by an off-the-shelf GEC system GECTOR (Omelianchuk et al., 2020), and predict their evidence words and error types using the interaction-based model with syntactic knowledge. We recruit 5 second-language learners as annotators to evaluate whether the predicted explanation is helpful in understanding the GEC corrections. The results show that 84.0 and 82.4 percent of the model prediction for gold GEC correction and GECTOR has explanations, and 87.9 and 84.5 percent of the explanations of EXPECT and gold GEC correction, respectively, are helpful for a language learner to understand the correction and correct the sentence. This show that the explainable GEC system trained on EXPECT can be used as a post-processing module for the current GEC system.

5.7 Case Study

We identify two phenomena from our syntactic and non-syntactic models based on labeling models:

Distant Words Identification. The non-syntactic model makes errors because it does not incorporate explicit syntactic modeling, particularly in long and complex sentences where it is difficult to identify distant evidence words. As shown in the first case of Figure 7, the non-syntactic model fails to consider evidence words, such as “apply”, that is located far away from the correction. However, the syntactic-based model is able to identify the evidence word “*apply*”.

Dependency Parsing Errors. Some evidence word identification errors are from the misleading parsing results in the long sentence (Ma et al., 2018). As shown in the second case of Figure 7, the model with syntactic knowledge is actually using an inaccurate parse tree in the green box from the off-the-shelf parser, which results in identifying redundant word “*off*”.

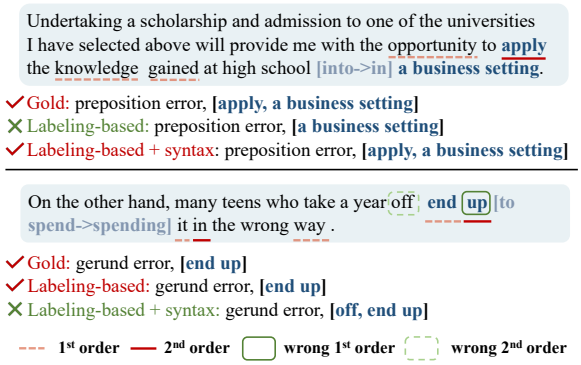

Figure 7: Case study. The first case shows the identification problem for distant evidence words. The second case shows the error caused by wrong dependency parsing results.

6 Conclusion

We introduce EXPECT, an explainable dataset for grammatical error correction, which contains 21,017 instances with evidence words and error categorization annotation. We implement several models and perform a detailed analysis to understand the dataset better. Experiments show that injecting syntactic knowledge can help models to boost their performance. Human evaluation verifies the explanations provided by the proposed explainable GEC systems are effective in helping second language learners understand the corrections. We hope that EXPECT facilitates future research on building explainable GEC systems.

Limitations

The limitations of our work can be viewed from two perspectives. Firstly, we have not thoroughly investigated seq2seq architectures for explainable GEC. Secondly, the current input of the explainable system is the gold correction during training, whereas, in practical applications, the input would be the output of a GEC system. We have not yet explored methods to bridge this gap.

Ethics Consideration

We annotate the proposed dataset based on W&I+LOCNESS, without copyright constraints for academic use. For human annotation (Section 3.3 and Section 5.6), we recruit our annotators from the linguistics departments of local universities through public advertisement with a specified pay rate. All of our annotators are senior undergraduate students or graduate students in linguistic majors who took this annotation as a part-time job. We pay them 60 CNY an hour. The local minimum salary in 2022 is 25.3 CNY per hour for part-time jobs. The annotation does not involve any personally sensitive information. The annotated is required to label factual information (i.e., evidence words inside the sentence.).

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. arXiv preprint arXiv:1910.02893.

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.

Christopher Bryant, Mariano Felice, and Edward Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. CoRR, abs/1611.01734.

Rod Ellis, Shawn Loewen, and Rosemary Erlam. 2006. Implicit and explicit corrective feedback and the acquisition of l2 grammar. Studies in second language acquisition, 28(2):339–368.

Simon Flachs, Felix Stahlberg, and Shankar Kumar. 2021. Data strategies for low-resource grammatical error correction. In Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications, pages 117–122.

David Freedman, Robert Pisani, and Roger Purves. 2007. Statistics (international student edition). Pisani, R. Purves, 4th edn. WW Norton & Company, New York.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. arXiv preprint arXiv:1803.07640.

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.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction. arXiv preprint arXiv:2203.07085.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. arXiv preprint arXiv:1911.00172.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard Hovy. 2018. Stack-pointer networks for dependency parsing. arXiv preprint arXiv:1805.01087.

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.

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.

Ryo Nagata, Kentaro Inui, and Shin'ichiro Ishikawa. 2020. Creating corpora for research in feedback comment generation. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 340–345.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task, pages 1–14.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: tag, not rewrite. arXiv preprint arXiv:2005.12592.

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.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. arXiv preprint arXiv:2106.03830.

Shuming Shi, Enbo Zhao, Bi Wei, Cai Deng, Leyang Cui, Xinting Huang, Haiyun Jiang, Duyu Tang, Kaiqiang Song, Wang Longyue, Chengyan Huang, Guoping Huang, Yan Wang, and Li Piji. 2023. Effidit: An assistant for improving writing efficiency.

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

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. In Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications, pages 37–47, Online. Association for Computational Linguistics.

Guihua Sun, Gao Cong, Xiaohua Liu, Chin-Yew Lin, and Ming Zhou. 2007. Mining sequential patterns and tree patterns to detect erroneous sentences. In AAAI, pages 925–930.

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022a. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. arXiv preprint arXiv:2203.13064.

Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022b. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 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.

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 EMNLP, pages 2518–2531.

A Appendix

A.1 Grammatical Error Categories

The definition of each grammatical error category in EXPECT is shown as follows:

- Infinitives: including errors like missing *to* before a certain verbs for to-infinitives, or unnecessary *to* after modal verbs for zero-infinitives.
- Gerund: misuse of the verb form that should act as a noun in a sentence.
- Participles: confuse with ordinary verbs like present simple, past simple or present continuous and other participles-related situations.
- Subject-verb agreement(SVA): the verb didn't agree with the number of the subject.
- Auxiliary verb: misuse of main auxiliary verbs like *do*, *have* or model auxiliary verbs like *could*, *may*, *should*, etc.
- Verb tense: incongruities in verb tenses, such as erroneous tense shift in a compound sentence, etc.
- Pronoun-antecedent agreement(PAA): pronouns didn't agree in number, person, and gender with their antecedents.
- Possessive: misuse of possessive adjectives and possessive nouns.
- Collocation: atypical word combinations that are grammatically acceptable but not common.
- Preposition: misuse of prepositional words.
- POS confusion: confusions in part of speech like noun/adjective confusion(e.g. difficulty, difficult), adjective/adverb confusion(e.g. ready, readily), etc.
- Article: wrong use of article.
- Number: confusion in singular or plural form of nouns.
- Transition: extra preposition after transitive verbs and missing proposition after intransitive verbs.

A.2 Implementation Details

We employ pre-trained `BERT-large-cased` in HuggingFace's Transformer Library (Wolf et al., 2020) as our encoder, which consists of 24 Transformer layers and 16 attention heads with 1024 hidden dimensions. We set the dimension of the correction embeddings and syntactic embeddings as 1024, which is the same as that in BERT. We set the learning rate to 1e-5 and batch size to 32 for non-interactive matrix models, and 5e-5 and 16 for interactive matrix models.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
After 6, Limitation section.

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
In abstract and section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

In 3,4,5 sections.

☑ B1. Did you cite the creators of artifacts you used?
In 3,4,5 sections.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
After 6, in ethics consideration section.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
After 6, in ethics consideration section.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
After 6, in ethics consideration section.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3

C ☑ Did you run computational experiments?

5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
In Appendix A.2 Implementation Details

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
In Appendix A.2 Implementation Details

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4, 5

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
3

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
3

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.